

Designing Game Feel: A Survey

Martin Pichlmair^{ID} and Mads Johansen^{ID}

Abstract—Game feel design is the intentional design of the affective impact of moment-to-moment interaction with games. In this article, we survey academic research and publications by practitioners to give a complete overview of the state of research concerning this aspect of game design. We analyzed over 200 sources and categorized their content according to design purposes. This resulted in three different domains of intended player experiences: 1) physicality; 2) amplification; and 3) support. In these domains, the act of polishing, which determines game feel, takes the shape of tuning, juicing, and streamlining, respectively. Tuning the physicality of game objects creates cohesion and predictability, and the resulting movement informs level design. Juicing is the act of polishing amplification and it results in empowerment and provides clarity of feedback. Streamlining allows a game to act on the intention of the player, supporting the execution of actions in the game. These three design intents are the main means through which designers control minute details of interactivity and inform the player’s reaction. The presented framework and its nuanced vocabulary can lead to an understanding of game feel that is shared between practitioners and researchers as highlighted in the concluding future research section.

Index Terms—Affect, feedback, game design, game feel, juice.

I. DEFINING OUR SCOPE

THIS article is a review of existing research about game feel. It uses reflections of practitioners and research publications for defining a more precise vocabulary to talk about the design of how interacting with a game feels.¹ The logical starting point for defining what “game feel” means is the book of the same name by Steve Swink, who defines it as “real-time control of virtual objects in a simulated space, with interactions emphasized by polish” [1]. He further expands on that definition by stating that great-feeling games convey five kinds of experiences, which are as follows:

- 1) the aesthetic sensation of control;
- 2) the pleasure of learning, practicing, and mastering a skill;
- 3) extension of the senses;
- 4) extension of identity;
- 5) interaction with a unique physical reality within the game.

Manuscript received November 12, 2020; revised March 10, 2021; accepted March 30, 2021. Date of publication April 9, 2021; date of current version June 16, 2022. (*Corresponding author: Martin Pichlmair*)

The authors are with the Center for Computer Games Research, Digital Design Department, IT University of Copenhagen, 2300 Copenhagen, Denmark (e-mail: mpic@itu.dk; madj@itu.dk).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TG.2021.3072241>.

Digital Object Identifier 10.1109/TG.2021.3072241

¹It is important to stress that we only refer to game examples in order to clarify design details but did not survey games.

While Swink’s definition of game feel covers a wide range of video games, it is too limited to encompass all kinds of them. With the aforementioned list, he excludes games without real-time controls from possessing the quality of game feel. Wilson [2] challenges this aspect of Swink’s book and extends the notion of “aesthetic sensation of control” by connecting game feel with the cultural history of gestures. Wilson distinguishes between “Game Feel” and “game feel,” the first being the positive feeling of control that Swink describes, the second being any feeling a game communicates. Jesse Schell does not mention the term “game feel” in his book [3] at all. Yet, he writes that designers should consider how their game feels in the context of required skills, learnability, and balance. The journalist and game maker Rogers wrote an exhaustive article [4] about what he calls “friction,” an alternative term for how a game feels. Friction is the experience of the player pushing against the boundaries of the system. It is the feeling of the inertia of the design working against the user’s force. Friction is often experienced by the player over a longer duration than moment-to-moment interaction, as it extends over several game elements. Friction can be the defining element of a game, or just a part of the experience. Rogers’ monolithic article is difficult to parse and, while detailed, does not go deep into specific aspects of game feel. It mostly recounts different feelings the author had during particular situations in games, establishing a wide vocabulary for talking about the aesthetic experience of playing. Similarly, Anthropy and Clark [5] establish a “game design vocabulary” in their book of the same name. In it, they approach friction from the designer’s perspective, and call it “resistance.” The resistance of the game determines the experienced friction. It decides how the game feels to the player.

Ehrndal [6] approaches the topic of game feel by linking reflections of practitioners with aesthetic theories of games. Larsen [7] starts from a similar point and attempts to define “aesthetics of action.” He builds on Swink [1] and Nijman [8], both game developers more than researchers, in order to analyze the components of a game that contribute to what he calls a “thrilling experience.” Yang [9], on the other hand, expands the theory of game feel to include the metaphorical aspects of game objects and their relations to players. Building on queer theory, he includes political aspects of games in their “feel” in order to communicate the diversity of the gameplay experience over diverse players, and also in order to provide game makers with a richer set of design tools.

While games are multisensory experiences, we are focusing on the haptic and visual aspects of game feel in this article, aware that narrative content, music, art, and many other aspects of a game influence how it feels. Very similar techniques to the ones

described in this article exist, for example, for designing the feel of the story of a game [10], writing the voices of in-game characters, balancing its rules, and tuning its atmosphere. We will leave those aspects and the social interaction between players to future research while acknowledging that they are at the core of a lot of games. Instead, this article is concerned with moment-to-moment interactivity [11]–[13], microinteractions [14], and interactions with core loops [15] and their design. Unlike Swink’s precise but narrow definition of “Game Feel,” we will look at game feel more broadly as the affective aspect of interactivity. Our focus, thereby, is on the player and the design of their interaction with the game.

II. PHYSICALITY OF INTERACTIVITY

This survey article gives an overview of the history, context, and state-of-the-art understanding of game feel and how to design it. It is based on research in the field and publications by practitioners in order to capture both conceptual and practical knowledge. This chapter gives an overview of academic lines of thinking that lead to an understanding of game feel.

A. From Flow to Feel

In the years before Swink wrote the book “Game Feel” in 2009 [1], research about the link between emotions and gameplay was most often connected to Csikszentmihalyi’s famous “Flow theory,” one of the results emerging from a global research project about experiences that are “so gratifying that people are willing to do it for its own sake, with little concern for what they will get out of it, even when it is difficult or dangerous” [16]. Tasks that allow for this quality of experience feature the following eight elements:

- 1) a task that can be completed;
- 2) the ability to concentrate on the task;
- 3) that concentration is possible because the task has clear goals;
- 4) that concentration is possible because the task provides immediate feedback;
- 5) the ability to exercise a sense of control over actions;
- 6) a deep but effortless involvement that removes awareness of the frustrations of everyday life;
- 7) concern for self disappears, but sense of self emerges stronger afterward;
- 8) the sense of the duration of time is altered.

The concept of Flow has had an enormous influence on the understanding of experiential qualities of games. Sweetser and Wyeth [17] adapted Flow theory to games and Juul [18] discusses and criticizes the theory’s relevance for describing enjoyable challenges in games. Both texts contain multiple references to how games make a player feel, so it is natural to think of them as stepping stones toward a closer examination of game feel. Ciccioricco [19] links Flow to the gameplay experience of Mirror’s Edge [20], a game that was sold on its merits in fluidity of movement, and contrasts it with the feminist concept of fluidity. Chen [21] famously not only based his graduation thesis on Flow but also released three successful commercial games based

on his understanding of this concept with Thatgamecompany, a studio he cofounded; Flow [22], Flower [23], and Journey [24].

Game feel is most strongly reliant on points three, four, and five in the aforementioned list of criteria. Clarity of goals will be discussed in the context of streamlining of the player experience in Section IV-C. Immediate feedback is at the heart of this article and of the link between a game and how it feels. The sense of control will be mentioned at various points, for example, in relation to the illusion of control and immersion. While some of the other items in Csikszentmihalyi’s list apply to games too, they do so in a more indirect way.

B. Purpose of Juice

While Flow is very well suited for understanding the dynamics of immersion, game feel is more focused on the role that interactivity plays in this process. A design concept often mentioned when talking about how interactivity can be intensified is “juice.” Juice amplifies interactivity by providing excessive amounts of feedback in relation to user input [25] (see also [26]). The goal of juice is to make actions feel significant and the results can be measured [27]. It is superfluous from a strictly mechanical perspective, but turns interacting with the system into a more pleasurable experience. Too much juice, though, makes it hard to learn what aspects of interactivity have mechanical importance [28]. Decoding the actual system behind the game becomes cumbersome when the whole screen is filled with wobbling particle effects—unless this is a conscious aesthetic choice and itself a part of the game’s mechanics.

At the same time, the diversity of the medium allows that some games—we can call them “toys” or “autotelic experiences” [29]—are almost purely made of juice. Interacting with those toys is still playful and based on feedback amplified by juice. Only through feedback can we learn (to play), and all play is learning (see [30]–[32]; see also [33] and [34]). One could even go so far as to argue that all cognition is rooted in feedback from the real world that we actively engage with in a process of interactive cognition (see [35], based on the work in [36]).

Overall, the goal of the application of juiciness is to enhance the feedback when interacting with game objects. Kao [37] conducted a large-scale study on the amount of juice appropriate for a specific gaming situation, concluding that juice has to be applied adequately to the situation. In their study, medium and high levels of juice outperformed extreme levels and the absence of juice across the measures of player experience, intrinsic motivation, play time, and in-game performance. Hicks *et al.* [26] bridge industry knowledge and academic analysis, building on Juul’s [25], [38], Schell’s [3], and Deterding’s [39] work on juice in video games (see also [40]). They present a framework for the analysis of “good game feel” in games.

Sometimes juice exists not for the player but for the audience watching the game. Rogers hints at that when he says “The player knows where the hit range of the weapon is. He doesn’t see the little juice-dance of the chain-daggers” [4]. Swink attributes a similar effect to ragdoll physics: “The ragdoll raptors have ‘over the shoulder’ appeal. People walking by someone playing

the game often stop and want to know more..." [1]. Gage [41] describes the upsides of having a game that is readable from distance in their talk on "subway legibility." More specifically, there are elements of juiciness that designers implement for the audience, especially for streaming and e-Sports [42]. Some elements might also draw in the player but become invisible to them over time.

Hunicke [43] remarks that "juiciness can be applied to abstract forms and elements and it is a way of embodying arbitrarily defined objects and giving them some aliveness, some *qua*, some thing, some tenderness." Interestingly, Swink [1], Larsen [7], as well as Fullerton [44] use the term "polishing" to describe something very similar to this. Fullerton describes the act of polishing as "the impression of physicality created by layering of reactive motion, proactive motion, sounds, and effects, and the synergy between those layers" [44]. In other words, she sees polish as a means of giving physicality to inanimate objects in order to render them more tangible, which is remarkably similar to Hunicke's reasoning for juiciness. Practitioners call many things "polish," e.g., fixing the timing of voice cues, or fixing bugs in the code (see, e.g., [45]). Polish is linked to juiciness in which all juicy elements are polished at some point, but it is seen as a mostly aesthetic endeavor that stops short of changing the basic rules of a game, its core narrative, or its principal game mechanics. In practice, this separation is not always maintained and the connection between juicing, polishing, designing, and the feelings elicited by the feedback loop of interacting with a game is complex.

C. Juicing That Feeling

The intentionality of polishing and juicing apparent in Hunicke's and Fullerton's comments is at the center of Brown's assertion that "you're not juicing your game—you're actually picking a feeling that your game should communicate and juicing that feeling" [46]. Feelings are emotions, but it would go too far to cover complex emotions like love, hate, and guilt in this article. The moment-to-moment interaction we are concerned with is instrumental to the feelings that Baumeister *et al.* [47] call "automatic affect." Automatic affect is closely linked to experience via feedback loops in which the affective reaction to a stimulus has an effect on future experiences of stimuli and those have an effect on the person experiencing them and so on. Emotions are generally characterized by physiological arousal, the quality of the experience, and valence, which can be either positive or negative [47]. Game designers are, of course, concerned with positive as well as negative emotions because stretches of sadness and near-frustrating challenges provide the perfect breeding ground for happiness and relief. It is important to note that humans are capable of experiencing multiple and even conflicting emotions simultaneously [48]. Furthermore, experiments in mood regulation have shown that humans exhibit a "homeostatic mood management mechanism" [49]. After initial mood-congruent responses, we spontaneously reverse and replace those by mood-incongruent reactions. So, in addition to the feedback between the outside world—including mediated

experiences like video games—and our emotional state, there is a feedback loop built into our mood.

These affective feedback loops are central to how we interact with the world, including the virtual world of video games. The connection between emotion and cognition is a vast research field and proponents of that field like Okon-Singer *et al.* often speak of how central emotions are to cognition [50]. The emotional aspect of design has been reflected by design thinkers, such as Löwgren, Kirkpatrick, Hodent, and Karhulati. Löwgren provides a vocabulary for linking aesthetics, design, and emotional responses [51], [52]. The sensibility and precision he employs to talk about design elements and design choices are valuable for better discussions about game design. Kirkpatrick [53]–[55] and Karhulahti [56] build an aesthetic theory of video games that encompasses kinesthetics as a foundational building block. Hodent [57] successfully bridges the chasm between interaction design and user experience design for games and accurately summarizes the links between Norman's work [58] and video game design. In general, user experience (UX) Design is an area that is concerned with the experiential aspects of interactivity within the vast field of human-computer interaction. Hassenzahl [59] presents an in-depth study of the complex links between needs, affect, and interactivity. Methods originating in UX design have found their way into games [60]. Their main influence is indirect. They inform the design and iteration process by offering a portfolio of tools and techniques. For example, Dan Saffer's proposal of microinteractions [14] links to game design in that the basic components of microinteractions are triggers, rules, feedback, and loops (or modes)—all basic building blocks of game design.

Playing is an embodied experience [61] that can be viewed through the lens of phenomenology. Keogh [62] argues that the phenomenology of play, rooted in the understanding of embodiment by Merleau-Ponty [63], Bateson [64], and Weiss [65], "must not start with the experience of the player's body, but with the experience through which the player's amalgam embodiment in and as part of the videogame performance emerges" [62]. Surman [66], Davnall [67], and Putney [68] offer three personal takes on three different games, echoing similar struggles of coming to terms with the bodily experience of playing games and the implications of the act of doing so. Sudnow [69] can be regarded as the trailblazer of writing about video games from a phenomenological standpoint.

In summary, game feel research is concerned with how our minds and our bodies experience the emotions of playing games. The question of how to design the emotional aspect of the play experience has been at the center of a lot of research that connects design theory, psychology, phenomenology, philosophy, and many more areas. The following section will go into the building blocks designers use to intentionally elicit emotions, when "juicing that feeling" [46].

III. GAME FEEL DESIGN ELEMENTS IN PRACTICE

Gameplay designers, of which some have a programming background and some have a design background, have analyzed their own practice in countless blog posts, podcast episodes,

conference presentations, and, sometimes, scholarly publications. The majority of works concerned with topics of game feel are descriptive in nature. They usually focus on either a single game or a specific feature or set of features that the designers have worked on. What can be learned from these texts, more than anything, is that experienced gameplay designers are very conscious about which aspects of their game are relevant for shaping its feel.

Some practitioners talk about how to structure game development processes around the design of game feel [70]–[72]. Others focus on giving broad overviews of techniques [8], [13], [73]–[75]. Podcasts and video series by experienced practitioners such as Eggplant: The Secret Lives of Games [76]–[79], The Clark Tank [80], and Game Maker’s Toolkit [72], [81], frequently discuss game feel design as a part of their coverage of game design topics.

Most practitioners focus on details and material aspects of design instead of connections between their work and the context they are working in. Noticeable exceptions are Hodent [57], who links game feel to classical concepts of game development such as the “3Cs” [82], User Experience Design, as well as to Norman’s theories on emotional design [58] and Song [83], who provides an excellent overview of how to model the feeling of impact in action games. Turner [84] wrote one of the few articles on how to influence game feel via sound design, based on his own work in game audio. Ismail [85] speaks about community development, explaining how communities of makers establish more and more sophisticated discourse about their practice over time. Yang [9], who has been mentioned before, always connects his personal practice to his academic research.

In general, the topics that these practical articles cover cannot easily be isolated from each other. They all concern feedback and how it relates to controls of a game. If the game is regarded as a feedback system (following [86]–[88]), then game feel can be seen as a modulation of the said feedback system. Designing game feel is designing the adequate feedback for eliciting a specific feeling or affective reaction. The following chapters list different design elements that determine the game feel, the feel of moment-to-moment interaction. We cluster design elements into classes according to the game’s discussed subsystem. Table I presents an overview of the areas we are looking at and lists the most relevant examples mentioned. The table is not an exhaustive overview of all aspects of game feel from a practitioner’s perspective. It is a starting point for understanding the practices that are most relevant, or most discussed, in regard to designing game feel.

A. Movement and Actions

The first category of design elements is concerned with movement of the character and other objects and with what happens if the character or an object collides. Controlling an on-screen character means navigating the game world and interacting with other characters and objects. Most writing on this aspect of game feel is concerned with 2-D games. Dahl and Kraus [89] provide a good starting point for exploring this topic. Normoyle and Jörg [90] look at the tradeoff between naturalness of movement and

TABLE I
GAME FEEL DESIGN ELEMENTS OVERVIEW

Design Element	Physicality	Amplification	Support	Key References
Movement and Actions				
Basic Movement	•			[89]–[95]
Gravity	•			[92], [96], [97]
Terminal Velocity			•	[92]
Coyote Time			•	[98], [99]
Invincibility Frames			•	[100]–[102]
Corner Correction			•	[28], [103]
Collision Shapes	•		•	[104]
Button Caching			•	[92]
Spring-locked Modes	•			[5], [105], [106]
Assisted Aiming			•	[28], [107], [108]
Event Signification				
Screen Shake	•	•	•	[8], [73], [109], [110]
Knock-back & Recoil	•	•		[8], [111]
One-shot Particle Effects		•	•	[73], [112]–[117]
Cooldown Visualisation			•	[118]–[121]
Ragdoll Physics	•	•	•	[1], [122]
Colour Flashing		•		[8], [71], [123]
Impact Markers		•	•	[83], [124]
Hit Stop		•	•	[81], [83], [125], [126]
Audio Feedback	•	•	•	[127]–[129]
Haptic Feedback	•	•	•	[83], [130]
Time Manipulation				
Freeze Frames		•	•	[83]
Slow Motion		•	•	[83]
Bullet Time		•	•	[131], [132]
Instant Replays		•	•	[83]
Persistence				
Trails			•	[112], [127], [133]
Decals & Debris			•	[127], [134]
Follow-Through	•			[135]
Fluid Interfaces	•		•	[136]–[138]
Idle Animations			•	[135], [139], [140]
Scene Framing				
Highlighting			•	[141]–[143]
Dynamic Camera		•	•	[71], [144]–[149]

responsiveness of controls. Pignole [91] describes ten different aspects of how to design controls that feel responsive. While purely grounded in his own experience, these recommendations are easy to pick up and adapt to any game with 2-D character movement. In a more extensive study, Fasterhold *et al.* [92] (see also [95]) provide an overview of parameters for modeling running and jumping in games. This article also contains an extensive literature review and insights into implementation details of various platformer games. The authors’ model features 21 different parameters to describe basic 2-D movement. The key argument in this article is that movement parameters afford [150] level patterns. Mario’s [151] jump curve, for example, excellently facilitates precision descents thanks to featuring terminal velocity that makes future positions easier to predict. *Super Meat Boy* [152], as another example, abruptly interrupts a



Fig. 1. *Super Meat Boy* allows the player to interrupt a jump, when the jump button is released, to avoid ceiling elements. Image from [92].

jump when the jump button is released, which makes hazardous ceiling elements a viable level design choice, since they can be avoided more easily than if the jump would continue. This is shown in Fig. 1.

Hamaluk [153] used screen recordings to measure and reconstruct all relevant parameters for *Super Mario World* [153]. Game Makers’ Toolkit (see [81]; see also [154]) runs a more informal analysis of the platformer *Celeste* [155]. *Celeste*’s player controller’s source code was published [156], allowing for even deeper analysis. Fiedler [157] provides good starting points for implementing advanced controls and simulations.

1) Basic Movement: This design element is concerned with the most basic parameters defining the interactive movement of an on-screen object, in most cases the player character. The parameters in question are speed, acceleration, friction, and breaking speed (see [89]–[91]). If the player can jump, the strength of the jump force as well as eventual air friction comes into play, too. In the case of 2-D games, Fasterhold *et al.* [92] list these and more parameters and how they are related. Saltsman [98] covers movement in one specific platformer, *Canabalt* [158], in greater detail. Pittman [93] explains the mathematics behind jump mechanics. The exact requirements for tuning the movement of a game is often so deeply connected to the gameplay that it is hard to generalize. An in-depth analysis of the car ball game hybrid *Rocket League* [159] is presented by Cone [94] and demonstrates how steering of a vehicle is tuned in similar ways to platformer movement.

2) Gravity: The strength of gravity defines how much force pushes an object toward the ground. Games rarely feature earthlike gravity, opting for higher values instead, in order to create a more controlled feeling. Fasterhold *et al.* [92] list the strength of gravity for a number of platformer games. Earth has a gravitational acceleration constant of 9.807 m/s^2 , whereas *Super Meat Boy*, assuming that the character is 1 m tall, has a constant of 41 m/s^2 and *Super Mario Bros.* even features 91.28 m/s^2 [96]. Gravity is used as a sophisticated game mechanic in *Super Mario Galaxy* [160], where the character can jump from planet to planet and always aligns appropriately with the surface of the cosmic body. Alessi [97] wrote up an explanation and prototypical implementation of this gravity system.

3) Terminal Velocity: The existence of terminal velocity in a movement system means that a falling object does not perpetually get faster. It stops to accelerate at a predefined speed, the terminal velocity. As mentioned earlier, Mario’s [151] jump curve [92] facilitates precision descends thanks to terminal



Fig. 2. Illustration of “Coyote Time” in *Canabalt* [158], showing the extra distance from the building where a jump is still accepted. Image from [98].

velocity. The additional predictability, which results from the curve becoming a line, supports precision.

4) Coyote Time: The term “Coyote Time” refers to a movement system that allows a player to still instigate a jump a short time span after running off a cliff.² It is perhaps the most famous example of supporting the intent of the player. A detailed account of its technical implementation in the minimal platforming game *Canabalt* [158] can be found in [98], shown in Fig. 2. Coyote Time is sometimes called “Coyote Jump” or “Ghost Jump” [99]. A similar time-based accessibility feature can be found in Disc Room [161], where hitboxes that kill the player get activated only after a delay of up to 50 ms.

5) Invincibility Frames: Short time spans where the player character is invincible. They are a side-effect of player actions such as rolling, dodging, respawning, or attacking. SmashPedia [100] lists 23 different cases of invincibility in Super Smash Bros. Ultimate [162], a fighting game, alone. These moments of invincibility are useful for normal players but essential to competitive play and speedrunning. Mora-Zamora and Brenes-Villalobos [101] describe invincibility frames as a tool for balancing risk and reward. Siu *et al.* [102] mention invincibility frames as a part of boss fights. The purpose for introducing a few frames of invincibility is usually to support the player to give them a carefully measured amount of safety that allows them to pull off even more spectacular actions than if they were vulnerable all the time.

6) Corner Correction: Adjusting a character’s path if it would otherwise get stuck in level geometry. This is a common convenience in games where walking is a large part of gameplay. Gilbert [103] analyzes how it is implemented in *The Legend of Zelda* [163] and Doucet [28] offers a detailed analysis of its implementation in *Super Mario Bros. 3* [151].

7) Collision Shapes: Collision detection is the problem of determining the intersections of “hitboxes” (also called “colliders”) of arbitrary shapes and, sometimes, preventing the interpenetration of these colliders to communicate that objects have physical extents in the world. In the case of a 2-D game, collision shapes are usually either circles, triangles, or rectangles. In 3-D games, they are often spheres, boxes, or capsules. The individual shapes and extents of hitboxes, as well as the coherence between

²The name “Coyote Time” is based on the coyote in the Road Runner series, a character who possesses the power to only fall from a cliff after realizing he had been running on thin air for a while.

collision shapes and visible game elements, determine how collisions between game elements feel to the player [104].

8) *Button Caching*: A common player support function is “Jump Buffering” [92], where the controller code buffers the pressing of the jump button for a few frames and executes the jump after the player has landed. Mario [164] caches the button for 1–2 frames and Braid [165] for 0.23 s [92].

9) *Spring-Locked Modes*: This is a user-interface modality that is actively maintained by the player by pressing and holding a button. The object they are interacting with “switches mode” for the duration that the button is held. This form of interaction is what Raskin [105] calls a “quasimode” and Johnson and Engelbeck [106] refer to as “spring-locked mode.” It is often used in order to create anticipation. Games where the player charges an action before unleashing it fall into this category (e.g., *Angry Birds* [166], *SSX Tricky* [167], and *R-Type* [168]). Exiting the mode can have a specific effect like the charged shot in *R-Type* (see [169]), or it just returns the player to the previous mode, like in the case of *Dark Souls* [170] where the player raises the shield by pushing a button and lowers it by lifting their finger again. Drag and drop is another example of a spring-locked mode that is common in game interfaces.

Furthermore, some games mirror the action of the player and the action of the character. Jumping in the snowboarding game *SSX Tricky* [167], for example, is charged by pressing a button and holding it. The character jumps at the moment when the button is released. This implementation creates a relation between the game mechanic and the physical action of the player [5] in which the bodily activity of the player releasing the button is more similar to the action of the controlled character than if the jump would be triggered by pressing the button.

10) *Assisted Aiming*: Some games help a player with the precision required for aiming. Many shooter games support assisted aiming (e.g., *Gears of War* [171]) and driving games come with countless driving assistance settings.³ These features can be regarded as manifestations of what Doucet calls “oil” [28], the measured exploitation of “illusion of control,” as discussed by Kayali and Purgathofer [107]. An extensive description of a particular case of assisted aiming for console shooters can be found in the work by Zimmerman [108].

B. Event Signification

This class of design elements signifies gameplay-relevant events. Similar techniques are used when events are triggered by the player and when they are triggered by the system. All techniques listed in this section are only active for a limited duration. It is usual to layer several effects depending on the significance and kind of event being communicated.

1) *Screen Shake*: This effect, which is sometimes also referred to as “camera shake,” shakes the camera (or the world) in order to communicate a significant event—often an explosion, taking damage, or similar high-impact actions. Nijman [8] and Jonasson and Purho [73] both mention screen shake. Lerpung

³DiRT 3 [172] features ABS, Dynamic Racing Line, Stability Control, Auto Steer, Corner Braking and Throttle Management—very similar systems can be found in real cars.

and easing functions [109], [110] form the technical basis of the implementation of dynamic cinematography, such as screen shake. Instead of randomly moving the camera, a carefully selected easing function in a semantically significant direction communicates more information about what has happened, giving the designer more control over what is communicated to the player.

2) *Recoil*: When the player character is slightly pushed back, e.g., after firing a gun. Nijman [8] describes an implementation in detail, where the firing of a bullet shakes the screen while also pushing the player character a few pixels back, resulting in a side-effect with gameplay implications. A more sophisticated way of achieving something similar is to use inverse kinematics. *God of War* [173] uses inverse kinematics to model the reaction of the body of the player character when catching his axe [111]. Not only the arm but the whole body of the character reacts.

3) *One-Shot Particles*: Particle systems [112]–[114] are a staple of juicy game design [73], [115]. Practitioners apply them according to context and sophisticated examples feature many layers of particles accompanied by other techniques from this list, such as screen shake and sound effects. Some simple particle systems can be faked using textures (see [8]). Rockenbeck [116] demonstrates a state-of-the-art particle pipeline and explains how it was used in *FAMOUS: Second Son* [174]. Vainio [117] describes how this particular system fits into the wider picture of a modern visual effect pipeline.

4) *Cooldown Visualization*: Cooldown time is the time it takes until an ability that was just triggered can be used again. Its visualization has to communicate how long the ability is unavailable as well as the moment it becomes available again. Cooldowns are mostly found in role-playing, where they govern how often spells can be cast or a character ability can be used, and in real-time strategy games, where they govern how long it takes to, e.g., research a technology. The duration of the cooldown is usually communicated by graying out the button that triggered an action and gradually revealing it again over the cooldown time. A short overview can be found in [118] and [119]. A detailed study of optimizing the display of cooldowns in a custom user interface can be found in [120]. Generally speaking, cooldown visualizations are a subset of progress indicators (see also [121]).

5) *Ragdoll Physics*: Modeling a character using joints, forces, and rigid bodies, instead of animations. Switching from animation to ragdoll is a staple for communicating that a character has died. Jakobsen [122] wrote about this design element before the name “ragdoll” became common. Swink describes [1] how they used ragdoll physics in the game *Off-Road Velociraptor Safari* [175], [176]. He also lists a number of games that derive their whole appeal from ragdoll physics.

6) *Color Flashing*: This simple but effective technique communicates state changes by overlaying an on-screen graphical object or parts of the screen with a color. Perry [71] mentions several practical examples of how to indicate damage or other state changes by, e.g., flashing the object color or flashing the whole screen. A special case is flashing an on-screen element that was destroyed before it gets removed from the screen, a technique that creates persistence over time, which is discussed in regards to several other design elements further down this

article. Research indicates that specific color choices carry different meanings [123]. Nijman [8] demonstrates flashing the enemy sprite white for a frame or two in a 2-D platformer to emphasize a hit.

7) *Impact Markers*: In the absence of a player character, for example, in first-person games, other visual elements have to be used to indicate events. In action games, especially in shooters, getting shot at is information of prime importance. Stephenson [124] lists several different techniques for signifying direction, kind, and strength of impact, illustrated by game examples. Song [83] explains a number of different elements, most of them covered in their own sections in this overview, specifically for signifying impact. A blend of the color flashing mentioned earlier and impact marking is, for example, achieved with impact lighting, where a light source gets created on impact that illuminates the characters from the point of impact [83].

8) *Hit Stop*: Animations pause for a brief moment on impact. This effect, sometimes also called “Impact Freeze,” is a staple in fighting and action games and maybe the best researched phenomenon in the area of impact feedback visualizations [83], [125], [126]. Brown [81] describes frame freezes and their design purposes in *Celeste* [155]. Hit stops are usually introduced in order to communicate feedback about the severity of a hit, but can go further than that. Samurai Gunn [177] features a subtle variation of impact freeze when a character lands on a platform, “stunning” it for a few frames depending on the height it dropped from. Kratos’ axe in *God of War* [173] freezes when it hits an enemy [111].

9) *Audio Feedback*: Acoustic channels of communication are a very common way of layering information on top of the graphical representation of a game. Additionally to supporting immersion, audio can also communicate events that happen off screen. Berbece [127] not only highlights the importance of sound effects but also explains how to layer several in order to create an easy-to-read soundscape. Audio feedback in interaction design for games can be regarded as a specific application of Sonic Interaction Design [128]. Nacke and Grimshaw [129] present research on affective and aesthetic impact of game sound. Overall, sound design is a huge part of game development and offers a rich set of tools and techniques (see, e.g., [178], [179]) that are relevant in relation to game feel but too general to cover in this article.

10) *Haptic Feedback*: Haptic feedback, often called “force feedback” or simply “controller vibration,” is a standard functionality of console controllers and built into most mobile phones. It is often used for emphasis rather than as a critical component to interactivity. Orozco *et al.* [130] provide a complete overview of the history and significance of haptic feedback for games. Most platform holders have clear guidelines about when to use haptic feedback, which means that platform-exclusive titles often exploit these features more than multiplatform games (see [83]).

C. Time Manipulation

While hitboxes and movement are spatial, the other dimension often exploited for game feel is time. Zagal and Mateas [180]

give a good overview of game time from an analytical standpoint. The design intent of time manipulation is most often to amplify the experience or to clarify the intensity or direction of an impact. In this section, game time refers to the time of the world simulated in the game, whereas real time refers to time in the real world.

All examples in this section have to do with slowing down or pausing game time because games only rarely speed up time to emphasize a moment. *SSX Tricky* [167] and *Bubble Bobble* [181] are among the few examples of games that do so.⁴ In *Drawkanoid* [182], a brick destruction game, time speeds up while the player is waiting for the ball to return from a brick’s destruction. No research about speeding up time has been found, so this chapter only covers the rest of the cases of time manipulation. Also left out from this section are rhythm and cadence of interactivity. While game designers are doubtlessly aware of the importance of those two aspects to game feel, there is not enough substantial writing to cover them.

1) *Freeze Frames*: The whole screen is frozen for a short duration, often just a few frames. The difference to hit stops, described earlier, is that those are a localized phenomenon where one or more on-screen objects get paused, excluding them from the temporal flow of the rest of the game, whereas freeze frames technically halt the progression of game time. Song [83] describes how some games pick the best frame of an animation to freeze on and what gameplay implications frame freezes have.

2) *Slow Motion*: Slowing down game time for a short duration. Whether applied to replays or to linear game time, slow motion helps to communicate events that would otherwise evolve too fast to be fully perceived by the player. A blend of impact freeze and slow motion can be found in *Holedown* [183]. The game does not fully freeze on impact, but slows down time to a near halt for a few frames instead. The ability to use slow motion to make an attack look more powerful is mentioned in the work by Song [83].

3) *Bullet Time*: Bullet time [131] is spring-locked slow motion. It serves as a way to pull off more spectacular or precise actions than the player could accomplish in real time. They empower the player, amplifying their actions. Porter [132] gives an overview of the history of bullet time in movies as well as games. Technically, bullet time is often eased in and out and maintained for a certain amount of time. This can be modeled using attack-decay-sustain-release “ADSR” curves (see [1] for details on their various applications).

Turn-based games such as *XCOM* [184] pause time while the player queues actions and subsequently advance it in order to show the results of these actions. *Fallout* [185] slows down time to a trickle in the V.A.T.S. mechanic, with very similar results. This pattern of letting a user plan a move without time pressure and then showing a lengthy and potentially intense payoff in real time is quite similar to the temporal dynamics of match-three games such as *Bejeweled* [186]. This particular way of manipulating game time could be regarded as an extreme form

⁴A lot of strategy and puzzle games allow users to set the overall speed of the game, which is not part of moment-to-moment interaction.



Fig. 3. Intense battle in *Samurai Gunn* [177], the history of motion and battle encoded in the white sword path, the bullet trajectory, smoke particles from where the gun was fired, as well as blood and gore traces all over the level.

of bullet time because it essentially fulfills the same purpose and has the same structure.

4) *Instant Replay*: While replays are technically taking control away from the player, and can as such be regarded as orthogonal to gameplay, they serve a similar purpose to juice insofar as they add clarity and weight to a gameplay situation. Replays of actions that have just happened [83], often slowed down, are usually triggered automatically. The design aspects of this technique, which originally comes from sports television, have not been researched in the context of games. The fact that replays communicate pivotal moments of gameplay means that replays might help players in identifying moments of importance. Interestingly, both games mentioned for their use of bullet time above, *XCOM* [184] and *Fallout* [185], feature slowed down replays.

D. Persistence

Another aspect related to time is persistence, which could also be called “temporal consistency.” Broadly speaking, this cluster of techniques uses spatial representation to communicate time-dependent information. The problem being solved is, in the words of Bay-Wei and Ungar: “When the user cannot visually track the changes occurring in the interface, the causal connection between the old state of the screen and the new state of the screen is not immediately clear.” [138].

From skid marks to particle trails, the purpose of the techniques listed ahead is always to encode information about the past in the currently displayed image. Even motion blur, which is mentioned further down the list in Section III-E, not only prevents temporal aliasing, but retains the history of movement as a lingering after-image. Very often, the below design elements are used in combination and additionally to other elements that communicate the dynamics of the on-screen action. An example of this can be seen in Fig. 3.

Temporal consistency also means a consistent frame rate. While this article neither covers technical details nor how bugs

and implementation weaknesses affect game feel, it is important to mention that frame rate and especially the duration of the physics time step have a huge influence on how reactive a game feels. Swink also stresses this when he maintains that “real-time control relies on sustaining three time thresholds: the impression of motion, perceived instantaneous response and continuity of response” [1]. Fiedler [187] provides an excellent introduction into how to implement a stable and reliable feeling core game loop. Cone [94] describes in depth how they solved countless challenges of running a stable physical simulation of the fast moving cars and ball in *Rocket League* [159].

Overall, temporal consistency techniques are employed in order to allow the player to see either past events or quickly happening events for a longer time. They can be implemented so that world space is used as an interface layer by attaching trails and particle effects to objects that would be invisible in the real world [188]. The key role of these techniques, from a design perspective, is to support the player.

1) *Trails*: Traces left behind a moving object. The most prevalent example of temporal persistence in games is found in particle systems [112] and trails. Particle systems that amplify the result of player interaction extend the time that result is visible on screen, creating a dynamic and, for a short while, persistent representation of the player’s interaction history, which could be seen as trails of a player interaction. Particle systems that leave a trail in space as well as time allow the reconstruction of the trajectory of movement of an object. These techniques increase the readability of a scene for the player as well as spectators [127], [133].

2) *Decals and Debris*: Decals and debris are stationary traces left in the game world. Birdwell [134] mentions how Valve used decals to acknowledge the actions of the player. Berbece [127] explains a specific design case, where the player character leaves a blotch of paint after being eliminated from a match, in great detail. Destructible levels are a special case of debris left by players that has physical implications and can be regarded as in-game level design.

3) *Follow-Through*: Follow-through, the effect where a part of an animated character or object keeps moving after the main motion has stopped [135] is also a way of encoding the history of the motion in subsequent frames. This time, the encoding is not done as a nondiegetic overlay or abstraction, but as movement of parts of the object in question.

4) *Fluid Interfaces*: Introduced by Apple in 2018 [136], “Fluid Interfaces” aim at offering more natural interaction forms based on aligning and understanding of intent with physical simulation. They aim at maintaining smooth continuity whenever possible. Gitter [137] summarizes the original presentation and provides a number of code examples. Continuity comes from temporal persistence and spatial coherence, for example, when a user-interface transition retains aspects of the previous view while transitioning to a new one [138]. In games, fluidity can be found in the moment when coins that are earned at the end of a round fly into a virtual purse, each with a short trail. Or when representations of pick-ups linger on screen after being collected and then attach to the character.



Fig. 4. Character camera-window in *Rastan Saga* [189], the camera only moves when Rastan pushes against this window. Image from [142].

5) *Idle Animations*: Small loops of animation that play after a while once the player stops interacting with their character—when the player character enters the “idle” state [139], [140]. They are superficial in relation to the core mechanics of the game but nevertheless contribute to the overall experience of a game. Idle animations are, of course, not triggered directly by the player. On the contrary, they are triggered indirectly by not interacting. Idle animations enhance the illusion of life [135] of the character.

E. Scene Framing

In racing and flying games, there is a tight link between the field of view, motion blur intensity, and speed. This link defines how the game feels. In 2-D games, a variety of techniques are used to enable specific game mechanics, support specific player behaviors, and give specific feedback to players. Keren [141], [142] assembled a great overview of these techniques. An example of a camera window can be seen in Fig. 4. Eiserloh [190] describes the effects of the maths behind camera controls on game feel.

1) *Highlighting*: Gameplay-relevant elements highlighted on and off screen. A lot of games have sophisticated techniques to direct the gaze of the player by gradually transitioning the camera focus from the player character to a point of interest. A good example of this is mentioned by Keren [141], [142] and explained in further detail by Meyer [143] on hand of his game *Insanely Twisted Shadow Planet* [191], an exploration game.

2) *Dynamic Camera*: Articles by Christie *et al.* [144], Haigh-Hutchinson [145]–[147], and Perry [71] provide good starting points for game camera design. Burelli [148] and Yannakakis *et al.* [149] examine affective reaction and camera handling. Burelli concludes that interactivity is the key difference between film cinematography and game cinematography, since his study “demonstrates how the impact on the player experience is mediated by her interaction” [148].

F. Summary

This list of elements of game feel design is by no means exhaustive. There are some obvious gaps stemming from how we scoped the review, e.g., narrative design, sound design, and

user-interface design. We hope that future researchers will use this list as a starting point for further exploration of the topic area. We are also aware that these techniques are differently suitable depending on genre, hardware platform, target group, and other factors of a game. Nevertheless, if a game designer concerns themselves with the aforementioned design elements and regards them as a collection of methods to draw from, they will be supported in intentionally conveying a chosen game feel.

IV. GAME FEEL

The classes of game feel design listed before are connected and, just like Jonasson and Purho keep adding juice in their talk [73], most moments of interacting with a game are shaped by the presence of several layers of feedback. A good example is the backstabbing attack in *Dark Souls* [170], an action role-playing game. This attack sequence is triggered by sneaking up to a foe from behind and attacking its back. If successful, the camera locks in place, the enemy and the player character get positioned in predetermined spots relative to each other, and weapon-dependent animations and sounds are played. The player is invincible for the duration of the sequence. The design purpose of this feedback set is to give weight to the effect of a single, but carefully prepared, button press. In general, game designers are most concerned with the quality of interactivity in the core loop [15] of a game, but that does not mean that they do not employ a lot of the techniques presented in this article in all parts of a game.

In the following paragraphs, we describe three design domains and what polishing in these domains entails. The domains are 1) physicality; 2) amplification; and 3) support. Physicality is the design domain describing the elements of the game that are a physical simulation. Amplification is the design domain concerned with intensifying the player experience. Support is the design domain where techniques that help the player to control the game are situated. Polishing means something different in each domain and this diversification helps us to talk about game feel design in a precise manner. Table II lists the three domains and their associated polishing task.

A. Tuning of Physicality

The first design domain is the experience of physicality of the system. Swink’s [1] whole concept of Game Feel rests on this pillar. Designers shape the feel of the game by tuning the parameters of the physical simulation [92]. Depending on how much the game’s core loop relies on the joy of movement, attention to detail can become extremely valuable [98]. Tuning of physicality leads to finely calibrated movement parameters, gravity, and collision shapes. The experience of control is enhanced by additionally applying screen shake, recoil, and knock-back. Appropriate audio design and haptic feedback additionally communicate the physical dynamics of gameplay. It is important to note that for the player, it often does not matter whether physicality is simulated or faked. Tweasing [192], specifically with easing functions [109], [110], and various other animation techniques [135], [193] can be used to communicate the desired weight of an object. These can be far easier to read as well as implement

TABLE II
GAME FEEL DESIGN DOMAINS AND THEIR ASSOCIATED POLISHING TASKS

Design Domain	Physicality	Amplification	Support
Domain Description	The physical simulation.	Intensification of experience.	Enabling the player.
Polishing Task	Tuning	Juicing	Streamlining
Task Description	Setting parameters to specify the behaviour of objects.	Adding feedback to emphasise, clarify and amplify.	Acting on player intent by interpreting the input in context of the gameplay situation.

than a realistic representation. Generally, tuning exploits our knowledge about physicality in order to make interactivity more predictable.

B. Juicing of Amplification

The second design domain is amplification. It primarily serves two purposes: First, it empowers the player. Second, it communicates the importance of events. Empowerment can take many shapes and forms. Bullet time, one of the most iconic ways of amplifying player actions, empowers the player to pull off more precise activities than they could if time progressed linearly. At the same time, it also signifies that the player has the opportunity to have greater impact during this time interval than during the rest of the game. Impact freeze, on the other hand, is mostly used to signify a successful interaction. Charging, which is based on spring-locking, is a technique that balances the reward of empowerment—the longer the player presses a button, the bigger the impact—with risk [194].

Juicing the amplification means providing adequate feedback to player actions and creating coherence between different aspects of feedback. Audio, haptic feedback, particle systems, and animation are the most important sources of juice. Juiciness requires exact timing of particle emissions, freeze frames, audio cues, perspective changes, and potentially many more parts of the game. Juicing empowers the player by structuring the reaction of the system to input in a way that amplifies actions adequately for the intended game feel.

C. Streamlining of Support

The third design domain is support. It covers techniques that help the player to execute a challenging action or just provide convenience. Doucet [28] calls the polishing of support mechanics “oiling,” whereas we adopt the less slick term “streamlining” that he also mentions in his article. Streamlining prevents player frustration by making sure that the player receives help where it supports the experience of the game. Doucet lists a couple of examples of how games can be streamlined in order to support the player. The goal of streamlining is to make rough edges of the game disappear, in order to provide a smooth player experience. Most of the time, the player does not want to realize how much the game is supporting them. “If you do this right then the player won’t suspect a thing” says Pulver [195]. *Disc Room’s* [161] designer Nijman explains that their use of Coyote Time “has a bunch of good side effects that make it seem like the game knows your intentions.” [104]. A large part of the 5400 lines of code that comprises the *Celeste* character controller is dedicated to providing forgiveness for the player (see [81] and [196] for an

overview of a few of the features implemented for this purpose). This results in controls that are “working on the player’s intent rather than making a precise simulation” [81]. A much more sophisticated approach was presented by Zimmerman [108] when they describe the selection mechanism for aiming targets when landing on the ground as one-dimensional optimization problem.

Elements that enhance temporal consistency offer a different kind of support. Trails that follow projectiles are an example of temporal feedback, since they help to determine the speed and direction of the object by documenting its history. Often particle systems in games have a similar role, and so do skid marks or trails in simulated mud or water. They serve as a visualization of the past and as an externalization of information that the player would otherwise have to memorize. That makes them a service for the player. Continuously displayed game elements like status effects and idle animations similarly lift the burden of remembering the state from the player’s shoulders by showing it in the game instead.

The act of polishing player supporting game elements is streamlining. Rather than explicitly informing the player about changes in the game’s state, streamlining is making the player’s experience adequately smooth. A good support system is often invisible [107]. User Experience Design [197], [198] can be used for this design area. A closely related cluster of research concerns game accessibility. While User Experience Design is concerned with setting up game development processes that encompasses user research, the role of accessibility is to widen the audience of games by providing guidelines and tools that make games accessible to players with different accessibility needs [199], [200].

D. Designing Game Feel

Game feel design is minute design work that evokes an affective reaction in the player. The precise reaction is subjective and highly dependent on context, inside and outside the game. Streamlining, tuning, and even juicing are techniques that help with consciously designing interactive challenges at the heart of the player experience. Game feel is a shortcut for describing how this experience feels. If game feel design is the act of fine-tuning the relationship between expected and actual outcome of an interactive process, then it must be regarded central to the game design process.

V. FUTURE RESEARCH

“Game feel” is a value-neutral expression. While game designers and scholars are mostly concerned with what they refer

to as “good game feel” (see, e.g., [26]), the subjective nature of game feel and the need for “good negative moments” [12] call for a more holistic terminology. What makes a negative moment “good” is a question for future research, especially in terms of how game feel affects players psychologically. In any case, those negative moments, if designed consciously, are a valid aesthetic choice, given that “aesthetics describes the desirable emotional responses evoked in the player, when she interacts with the game system” [88]. Since game feel is the experience of a game’s aesthetic—following Hunnicke’s use of that term—it spans visual elements, sound design, mechanics, as well as storytelling aspects. Continued exploration of these different game elements and how they contribute to game feel is a worthwhile research endeavor. Sound design, encompassing both sound effects and music, stands out as a field, which demands further research. Isolating the aspects that represent polish in different design domains would also be very worthwhile, for example, in narrative design, physics simulation, and animation. Applying this research to specific technological areas like virtual and augmented reality could also prove worthwhile. Social aspects and historical perspectives could also yield interesting results, especially if linked to related areas of design beyond digital entertainment, such as board games, physical play, sports, and other leisure activities.

Historical trends and useful insights could be gleaned from case studies that dig deep into various aspects of game feel across genres and over time. Another interesting observation we had while writing this article is that game feel elements often replace senses that are not part of the vocabulary of the computer. For example, inertia, smell, and the balance of your own body are not directly parts of the game. The methods described in this article are often used as a substitute for them.

Another potential area for future research is game design tools. Recent advances have led to tools for designing feedback [74], [75] and generating it [201], [202]. AI agents and algorithms to help designers analyze and adjust game difficulty [203], [204] and the flow of game play and levels [205]–[207] have been explored academically. Tools that support automatic game design [208], [209] may also benefit from research into feedback readability and assistance in creating the right feedback. Systems that either partially or fully generate games could benefit from being able to evaluate whether different parts of a game fit together (see also [5]).

Research on the effect of feedback on the readability of a game by an autonomous agent is sparse for now. Most of the general video game playing research [210]–[214] is powered by the Video Game Description Language [210], [215], where feedback is almost nonexistent. If game feel can provide support to human players, it might also be able to help AI agents.

Designers often work on game feel in very intuitive ways, and literature about game feel is very domain-specific. Most written records are by practitioners who are discussing their own projects. There are large areas that have not been reflected upon. Some of these relate to unusual game mechanics, for example, speeding up time. Others are of more general relevance, for example, the link between audio design and game feel. The purpose of this article is to give an overview of existing research

and techniques in order to make game feel more accessible to game designers and researchers.

Ultimately, we hope this article stimulates the creation of more nuanced and reflective design processes, the development of better design tools, and improved game design. If, as Keogh [216] puts it, “Mechanics are the skeleton. ‘Polish’ or ‘feel’ or ‘juice’ is the meat,” then a more precise vocabulary is a step toward cooking up better games.

ACKNOWLEDGMENT

The authors would like to thank Sebastian Risi, Hans-Joachim Backe, Dom Ford, Karin Ryding, Sílvia Fornós, Christian Hviid Mortensen, Charlene Putney, Miruna Vozaru, and Miguel Sicart for their support, feedback, proof reading, and inspiration. The authors would also like to thank Mike Cook, Martin Jonasson, and Steve Swink for great discussions about game feel. Finally, the authors want to bow their heads to Petri Purho and Jan Willem Nijman for having great (and ultimately similar) ideas about game feel and continuously talking about them.

REFERENCES

- [1] S. Swink, *Game Feel*. San Mateo, CA, USA: Morgan Kaufmann, 2009.
- [2] D. Wilson, *A Tale of Two Jousts: Multimedia, Game Feel, and Imagination*, 2016. [Online]. Available: <https://www.youtube.com/watch?v=hpdcekk4hLA8>
- [3] J. Schell, *The Art of Game Design: A Book of Lenses*. San Francisco, CA, USA: Morgan Kaufmann, 2008.
- [4] T. Rogers, “In Praise of Sticky Friction,” Kotaku. Accessed: Apr. 22, 2021. [Online]. Available: <https://kotaku.com/in-praise-of-sticky-friction-5558166>
- [5] A. Anthropy and N. Clark, *A Game Design Vocabulary: Exploring the Foundational Principles Behind Good Game Design*, 1st ed. Reading, MA, USA: Addison-Wesley, 2014.
- [6] M. Ehrndal, “A holistic approach to designing for a specific aesthetic experience in digital games,” M.S. thesis, Dept. Faculty Culture Soc., Malmö Högskola, Malmö, Sweden, 2012. [Online]. Available: <http://muep.mau.se/handle/2043/13942>
- [7] L. J. Larsen, “Collision thrills: Unpacking the aesthetics of action in computer games,” *J. Comput. Games Commun.*, vol. 1, no. 1, pp. 41–52, Apr. 2016.
- [8] J. W. Nijman, “Netherlands, The art of screenshake,” 2013. Accessed: Apr. 22, 2021. [Online]. Available: <https://www.youtube.com/watch?v=AJdEqssNZ-U>
- [9] R. Yang, “Queer futures in game feel,” Accessed: Apr. 22, 2021. [Online]. Available: <https://www.blog.radiator.debacle.us/2018/10/queer-futures-in-game-feel.html>
- [10] K. Vonnegut, *The Shapes of Stories*, 1985. [Online]. Available: <https://www.youtube.com/watch?v=oP3c1h8v2ZQ>
- [11] S. Kumari, S. Deterding, and J. Freeman, “The role of uncertainty in moment-to-moment player motivation: A grounded theory,” in *Proc. Annu. Symp. Comput.-Human Interact. Play*, 2019, pp. 351–363.
- [12] S. Sivak, “GAME 3400 level design - Moment based design.” 2012. [Online]. Available: <https://www.slideshare.net/sjsivak/game-3400-level-design-moment-based-design>
- [13] S. Swink, “Game feel: The secret ingredient,” 2007. [Online]. Available: https://www.gamasutra.com/view/feature/130734/game_feel_the_secret_ingredient.php?print=1
- [14] D. Saffer, *Microinteractions: Designing With Details*. Newton, MA, USA: Reilly Media, Inc., 2013.
- [15] M. A. Sicart, “Loops and metagames: Understanding game design structures,” in *Proc. 10th Int. Conf. Found. Digit. Games*, Jun. 22–25, 2015, pp. 1–9.
- [16] M. Csikszentmihalyi, *Flow: The Psychology of Optimal Experience*. New York, NY, USA: Harper and Row, 1990.
- [17] P. Sweetser and P. Wyeth, “GameFlow: A model for evaluating player enjoyment in games,” *Comput. Entertainment*, vol. 3, no. 3, pp. 1–24, Jul. 2005.

- [18] J. Juul, *Half-Real: Video Games Between Real Rules and Fictional Worlds*. Cambridge, MA, USA: MIT Press, 2005.
- [19] D. Ciccoricco, "Narrative, cognition, and the flow of Mirror's Edge," *Games Culture*, vol. 7, pp. 263–280, Jul. 2012. [Online]. Available: <https://journals.sagepub.com/doi/10.1177/1555412012454223>
- [20] "Mirror's Edge," DICE, Stockholm, Sweden, 2008.
- [21] J. Chen, "Flow in games (and everything else)," *Commun. ACM*, vol. 50, no. 4, pp. 31–34, Apr. 2007. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1232743.1232769>
- [22] "Flow," Thatgamecompany, Santa Monica, CA, USA, 2006.
- [23] "Flower," Thatgamecompany, Santa Monica, CA, USA, 2009.
- [24] "Journey," Thatgamecompany, Santa Monica, CA, USA, 2012.
- [25] J. Juul and J. S. Begy, "Good feedback for bad players? A preliminary study of 'juicy' interface feedback," in *Proc. 1st Joint FDG/DiGRA Conf.*, 2016, p. 1. [Online]. Available: <https://www.jesperjuul.net/text/juiciness.pdf>
- [26] K. Hicks, P. Dickinson, J. Holopainen, and K. Gerling, "Good game feel: An empirically grounded framework for juicy design," in *Proc. DiGRA Int. Conf., Game Message*, Jul. 2018, pp. 1–17. [Online]. Available: http://www.digra.org/wp-content/uploads/digital-library/DIGRA_2018_Paper_35.pdf
- [27] K. Hicks, T. Pike, K. Gerling, O. Burman, G. Richardson, and P. Dickinson, "Understanding the effects of gamification and juiciness on players," in *Proc. IEEE Conf. Games*, 2019, pp. 1–8.
- [28] L. Doucet, "Oil it or spoil it!" Aug. 2016. [Online]. Available: <https://www.fortressofdoors.com/oil-it-or-spoil-it/>
- [29] M. Sicart, *Play Matters*. Cambridge, MA, USA: MIT Press, 2014.
- [30] J. P. Gee, *Good Video Games Good Learning: Collected Essays on Video Games, Learning, and Literacy*. Bern, Switzerland: Peter Lang, 2007.
- [31] R. Koster, *A Theory of Fun for Game Design*. Scottsdale, AZ, USA: Paraglyph Press, 2005.
- [32] B. Sutton-Smith, *The Ambiguity of Play*. Cambridge, MA, USA: Harvard Univ. Press, 1997.
- [33] I. Iacovides, A. L. Cox, P. McAndrew, J. Aczel, and E. Scanlon, "Game-play breakdowns and breakthroughs: Exploring the relationship between action, understanding, and involvement," *Human-Comput. Interact.*, vol. 30, no. 3/4, pp. 202–231, May 2015.
- [34] M. Pichlmair, "Designing for emotions: Arguments for an emphasis on affect in design," Ph.D. dissertation, Dept. Faculty Comput. Sci., Vienna Univ. Technol., Vienna, Austria, 2004.
- [35] H. Gedenryd, "How designers work - Making sense of authentic cognitive activities," *Lund Univ. Cogn. Stud.*, vol. 75, pp. 1–123, 1998.
- [36] J. Dewey, "The quest for certainty: A study of the relation of knowledge and action," *J. Philos.*, vol. 27, no. 1, pp. 14–25, 1930.
- [37] D. Kao, "The effects of juiciness in an action RPG," *Entertainment Comput.*, vol. 34, Feb. 2020, Art. no. 100359.
- [38] J. Juul, *A Casual Revolution*. Cambridge, MA, USA: MIT Press, 2009.
- [39] S. Deterding, "The lens of intrinsic skill atoms: A method for gameful design," *Human-Comput. Interact.*, vol. 30, no. 3/4, pp. 294–335, May 2015.
- [40] S. Atanasov, "Juiciness: Exploring and designing around experience of feedback in video games," M.S. thesis, Dept. Faculty Culture Soc. (KS), Malmö Högskola, Malmö, Sweden, 2013. [Online]. Available: <http://mup.mau.se/handle/2043/15692>
- [41] Z. Gage, *Building Games That Can Be Understood at a Glance*, 2018. [Online]. Available: https://www.youtube.com/watch?v=YISKcRDCDJg&ab_channel=GDC
- [42] C. Carlsson and A. Pelling, "Designing spectator interfaces for competitive video games," M.S. thesis, Chalmers Univ. Technol., Gothenburg, Sweden, 2015. [Online]. Available: <http://publications.lib.chalmers.se/records/fulltext/224247/224247.pdf>
- [43] R. Hunnicke, "Loving your player with juicy feedback," dConstruct 2009, 2009. [Online]. Available: <http://2009.dconstruct.org/podcast/juicyfeedback>
- [44] T. Fullerton, *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*, 3rd ed. Boca Raton, FL, USA: CRC Press, Apr. 2014.
- [45] P. Suddaby, "5 important ways to add polish to your game," May 2013. [Online]. Available: <https://gamedevelopment.tutsplus.com/articles/5-important-ways-to-add-polish-to-your-game--gamedev-7642>
- [46] L. Brown, *The Nuance of Juice*, Vector, 2016. [Online]. Available: <https://www.youtube.com/watch?v=qtgWBUOjk4>
- [47] R. F. Baumeister, K. D. Vohs, C. N. DeWall, and L. Zhang, "How emotion shapes behavior: Feedback, anticipation, and reflection, rather than direct causation," *Personality Social Psychol. Rev.*, vol. 11, no. 2, pp. 167–203, May 2007.
- [48] J. T. Larsen, A. P. McGraw, and J. T. Cacioppo, "Can people feel happy and sad at the same time?" *J. Personality Social Psychol.*, vol. 81, no. 4, pp. 684–696, 2001.
- [49] J. P. Forgas and J. V. Ciarrochi, "On managing moods: Evidence for the role of homeostatic cognitive strategies in affect regulation," *Personality Social Psychol. Bull.*, vol. 28, no. 3, pp. 336–345, Mar. 2002.
- [50] H. Okon-Singer, T. Hendler, L. Pessoa, and A. J. Shackman, "The neurobiology of emotion cognition interactions: Fundamental questions and strategies for future research," *Front. Human Neurosci.*, vol. 9, Feb. 2015, Art. no. 58.
- [51] J. Löwgren, "Pliability as an experiential quality: Exploring the aesthetics of interaction design," *Artifact, J. Des. Pract.*, vol. 1, no. 2, pp. 85–95, 2007.
- [52] J. Löwgren, "Toward an articulation of interaction esthetics," *New Rev. Hypermedia Multimedia*, vol. 15, no. 2, pp. 129–146, Aug. 2009.
- [53] G. Kirkpatrick, "Between art and gameness: Critical theory and computer game aesthetics," *Thesis Eleven*, vol. 89, no. 1, pp. 74–93, May 2007, doi: [10.1177/0725513607076134](https://doi.org/10.1177/0725513607076134).
- [54] G. Kirkpatrick, "controller, hand, screen: Aesthetic form in the computer game," *Games Culture*, vol. 4, no. 2, pp. 127–143, Apr. 2009.
- [55] G. Kirkpatrick, *Aesthetic Theory and the Video Game*. New York, NY, USA: Manchester Univ. Press, 2011.
- [56] V.-M. Karhulahti, "A kinesthetic theory of videogames: Time-critical challenge and aporetic rhematic," *Game Stud.*, vol. 13, no. 1, p. 1, 2013.
- [57] C. Hodent, "Skill-building series: Emotion in game design (A UX Perspective)," 2020. [Online]. Available: <https://www.gdcvault.com/play/1026790/Skill-Building-Series-Emotion-in>
- [58] D. A. Norman, *Emotional Design: Why We Love (or hate) Everyday Things*. New York, NY, USA: Basic Books, 2005.
- [59] M. Hassenzahl, S. Diefenbach, and A. Göritz, "Needs, affect, and interactive products of user experience," *Interacting Comput.*, vol. 22, no. 5, pp. 353–362, 2010.
- [60] S. Long, "What is games user experience (UX) and how does it help?" Oct. 17. [Online]. Available: https://www.gamasutra.com/blogs/SebastianLong/20171002/306649/What_Is_Games_User_Experience_UX_and_How_Does_It_Help.php
- [61] K. Spiel and K. Gerling, "The surrogate body in play," in *Proc. Annu. Symp. Comput.-Human Interact. Play*, 2019, pp. 397–411.
- [62] B. Keogh, *A Play of Bodies: How We Perceive Videogames*. Cambridge, MA, USA: MIT Press, 2018.
- [63] M. Merleau-Ponty, *Phenomenology of Perception*. Evanston, IL, USA: Routledge, 1982.
- [64] G. Bateson, *Steps to an Ecology of Mind: Collected Essays in Anthropology, Psychiatry, Evolution, and Epistemology*. Chicago, IL, USA: Univ. Chicago Press, 1972.
- [65] G. Weiss, *Body Images: Embodiment as Intercorporeality*. New York, NY, USA: Routledge, 1999.
- [66] D. Surman, "Pleasure, spectacle and reward in Capcom's Street Fighter series David Surman," Videogame, player, text, pp. 204–221, 2007.
- [67] B. Davnall, "Dr Johnson's sore toe: Touch, naturalism and kingdom hearts," Sep. 2016. [Online]. Available: <http://startswithafish.blogspot.com/2016/09/dr-johnsons-sore-toe-touch-naturalism.html>
- [68] C. Putney, "Praise the sun: On yoga and dark souls," May 2016. [Online]. Available: <http://alphachar.com/praisethesun>
- [69] D. Sudnow, *Pilgrim in the Microworld*. New York, NY, USA: Warner Books, 1983.
- [70] K. Gray, K. Gabler, S. Shodhan, and M. Kunic, "How to prototype a game in under 7 days," 2005. [Online]. Available: https://www.gamasutra.com/view/feature/130848/how_to_prototype_a_game_in_under_7_.php
- [71] L. Perry, "The single most useful advice I can give for making any game better feedback," 2013. [Online]. Available: https://gamasutra.com/blogs/LeePerry/20130506/191739/The_single_most_useful_advice_I_can_give_for_making_any_game_better_feedback.php
- [72] M. Brown, *Secrets of Game Feel and Juice*, 2015. [Online]. Available: https://www.youtube.com/watch?v=216_5nu4aVQ
- [73] M. Jonasson and P. Purho, *Juice It or Lose It*, 2012. [Online]. Available: <https://www.youtube.com/watch?v=Fy0aCDmgmxg>
- [74] R. Foresti, *Best Practices for Fast Game Design in Unity*, Unite LA 2018, 2018. [Online]. Available: <https://www.youtube.com/watch?v=NU29QKag8a0>

- [75] R. Forestié, *How to Design With Feedback and Game Feel in Mind - Shake It 'Til You Make It*, Unite Copenhagen 2019, 2019. [Online]. Available: <https://www.youtube.com/watch?v=yCKI9T3sSv0>
- [76] N. Suttner, A. Nealen, Z. Gage, and D. Wilson, “Eggplant: The secret lives of games (Formerly the Spelunky Showlike) 42: The secrets of simplicity with Martin Jonasson,” 2020. [Online]. Available: <https://thespelunkishowlike.libsyn.com/42-the-secrets-of-simplicity-with-martin-jonasson>
- [77] N. Suttner, A. Nealen, Z. Gage, and D. Wilson, “Eggplant: The secret lives of games (Formerly the Spelunky Showlike) 39: The tricks of the toolkit with Mark Brown,” 2020. [Online]. Available: <https://thespelunkishowlike.libsyn.com/39-gmtk>
- [78] N. Suttner, A. Nealen, Z. Gage, and D. Wilson, “Eggplant: The secret lives of games (Formerly the Spelunky Showlike) 38: The rhythms and layers of Ryan Clark,” 2020. [Online]. Available: <https://thespelunkishowlike.libsyn.com/38-the-rhythms-and-layers-of-ryan-clark>
- [79] N. Suttner, A. Nealen, Z. Gage, and D. Wilson, “Eggplant: The secret lives of games (Formerly the Spelunky Showlike) 36: Game feel as procrastination with Jan Willem Nijman,” 2020. [Online]. Available: <https://thespelunkishowlike.libsyn.com/36-game-feel-as-procrastination-with-jan-willem-nijman>
- [80] R. Clark, “The Clark Tank,” 2019.
- [81] M. Brown, *Why Does Celeste Feel so Good to Play?| Game Maker’s Toolkit*, 2019. [Online]. Available: <https://www.youtube.com/watch?v=yorTG9at90g>
- [82] C. McEntee, “Rayman origins,” *Game Developer Magazine*, pp. 26–31, 2012.
- [83] J. Song, “Improving the combat “impact” of action games,” Gamasutra. Accessed: Apr. 22, 2021. [Online]. Available: https://www.gamasutra.com/view/feature/130703/improving_the_combat_impact_of_.php
- [84] J. Turner, “Oh my! That sound made the game feel better!” 2015. [Online]. Available: <https://www.gdcvault.com/play/1022808/Oh-My-That-Sound-Made>
- [85] R. Ismail, “Six stages of game dev community development,” 2015. [Online]. Available: https://www.gamasutra.com/blogs/RamiIsmail/20150504/242486/Six_stages_of_game_dev_community_development.php
- [86] D. Cook, “What are game mechanics?” Oct. 2006. [Online]. Available: <https://lostgarden.home.blog/2006/10/24/what-are-game-mechanics/>
- [87] D. Cook, “Loops and arcs,” Apr. 2012. [Online]. Available: <https://lostgarden.home.blog/2012/04/30/loops-and-arcs/>
- [88] R. Hunnicke, M. LeBlanc, and R. Zubek, “MDA: A formal approach to game design and game research,” in *Proc. AAAI Workshop Challenges Game AI*, vol. 4, May 2004, Art. no. 1722.
- [89] G. Dahl and M. Kraus, “Measuring how game feel is influenced by the player avatar’s acceleration and deceleration: Using a 2D platformer to describe players’ perception of controls in videogames,” in *Proc. 19th Int. Academic Mindtrek Conf.*, 2015, pp. 41–46.
- [90] A. Normoyle and S. Jörg, “Trade-offs between responsiveness and naturalness for player characters,” in *Proc. 7th Int. Conf. Motion Games*, 2014, pp. 61–70.
- [91] Y. Pignole, “Platformer controls: How to avoid limpness and rigidity feelings,” 2014. [Online]. Available: https://www.gamasutra.com/blogs/YoannPignole/20140103/207987/Platformer_controls_how_to_avoid_limpness_and_rigidityFeelings.php
- [92] M. Fasterholdt, M. Pichlmair, and C. Holmgård, “You say jump, I say how high? Operationalising the game feel of jumping,” in *Proc. 1st Int. Joint Conf. DiGRA FDG*, 2016, pp. 1–16. [Online]. Available: http://www.digra.org/wp-content/uploads/digital-library/paper_248.pdf
- [93] K. Pittman, *Math for Game Programmers: Building a Better Jump*, 2016. [Online]. Available: https://www.youtube.com/watch?v=hG9SzQxaCm8&ab_channel=GDC
- [94] J. Cone, “It is rocket science! The physics of ‘Rocket League’ detailed,” 2018. [Online]. Available: <https://www.gdcvault.com/play/1025341/It-IS-Rocket-Science-The>
- [95] A. Summerville, J. Osborn, C. Holmgård, and D. W. Zhang, “Mechanics automatically recognized via interactive observation: Jumping,” in *Proc. 12th Int. Conf. Found. Digit. Games*, 2017, pp. 1–10.
- [96] A. Lefky and A. Gindin, “Acceleration due to gravity: Super Mario Brothers,” *The Physics Factbook*, Accessed: Apr. 22, 2021. [Online]. Available: <https://hypertextbook.com/facts/2007/mariogravity.shtml>
- [97] J. Alessi, “Games Demystified: Super Mario Galaxy,” 2008. [Online]. Available: https://www.gamasutra.com/view/feature/131997/games_demystified_super_mario_.php
- [98] A. Saltsman, “Tuning Canabalt,” 2010. [Online]. Available: https://www.gamasutra.com/blogs/AdamSaltsman/20100929/88155/Tuning_Canabalt.php
- [99] M. Venturelli, “Game feel tips I: The Ghost Jump,” 2014. [Online]. Available: https://gamasutra.com/blogs/MarkVenturelli/20140810/223001/Game_Feel_Tips_I_The_Ghost_Jump.php
- [100] Smashpedia, “Invincibility frame.” [Online]. Available: <https://bit.ly/2JEPrgr>
- [101] R. Mora-Zamora and E. Brenes-Villalobos, “Integrated framework for game design,” in *Proc. 9th Latin Amer. Conf. Human Comput. Interact.*, Sep. 2019, pp. 1–6.
- [102] K. Siu, E. Butler, and A. Zook, “A programming model for boss encounters in 2D action games,” *Exp. AI Games: Papers AIIDE Workshop*, AAAI, Reston, VA, USA, Tech. Rep. WS-16-22, 2016.
- [103] T. Gilbert, “Movement mechanics,” 2012. [Online]. Available: <https://troygilbert.com/deconstructing-zelda/movement-mechanics/>
- [104] A. Wiltshire, “How hitboxes work,” Aug. 2020. [Online]. Available: <https://www.pegamer.com/how-hitboxes-work/>
- [105] J. Raskin, *The Humane Interface: New Directions for Designing Interactive Systems*. New York, NY, USA: ACM Press, 2000.
- [106] J. Johnson and G. Engelbeck, “Modes survey results,” *SIGCHI Bull.*, vol. 20, no. 4, pp. 38–50, Apr. 1989.
- [107] F. Kayali and P. Purgathofer, “Two halves of play-simulation versus abstraction and transformation in sports videogames design,” *Eludamos, J. Comput. Game Culture*, vol. 2, no. 1, pp. 105–127, 2008.
- [108] C. Zimmerman, “Reading the player’s mind through his thumbs: Inferring player intent through controller input,” 2010. [Online]. Available: <https://www.gdcvault.com/play/1012339/Reading-the-Player-s-Mind>
- [109] R. Penner, *Robert Penner’s Programming Macromedia Flash MX*. New York, NY, USA: McGraw-Hill, 2002.
- [110] A. Sitnik and I. Solovej, “Easing functions cheat sheet.” Accessed: Apr. 5, 2020. [Online]. Available: <http://easings.net/>
- [111] C. Barlog, *Why Kratos’ Axe Feels so Powerful|Game Mechanics Explained*, May 2018. [Online]. Available: https://www.youtube.com/watch?v=zpr-EE2In1M&ab_channel=Polygon
- [112] W. T. Reeves, “Particle systems a technique for modeling a class of fuzzy objects,” *ACM Trans. Graph.*, vol. 2, no. 2, pp. 359–375, Apr. 1983. [Online]. Available: <https://www.lri.fr/~mbl/ENS/IG2/devoir2/files/docs/fuzzyParticles.pdf>
- [113] T. Ilmonen and J. Kontkanen, “The second order particle system,” *J. WSCG*, vol. 11, no. 2, pp. 240–247, 2003.
- [114] L. Latta, “Building a million-particle system,” 2004. [Online]. Available: https://www.gamasutra.com/view/feature/130535/building_a_millionparticle_system.php
- [115] N. Lovato, “Squeezing more juice out of your game design!” Mar. 2015. [Online]. Available: <https://bit.ly/2GsjSex>
- [116] B. Rockenbeck, “The inFAMOUS: Second son particle system architecture,” 2014. [Online]. Available: <https://www.gdcvault.com/play/1020367/The-inFAMOUS-Second-Son-Particle>
- [117] M. Vainio, *The Visual Effects of inFAMOUS: Second Son*, 2014. [Online]. Available: https://www.youtube.com/watch?v=o2yFxPY2b1o&ab_channel=GDC
- [118] “Cooldowns can be used to balance games.” [Online]. Available: https://game-design-snacks.fandom.com/wiki/Cooldowns_can_be_used_to_balance_games
- [119] J. Griesemer, “Design by numbers: Cooldowns,” Jan. 2012. [Online]. Available: <https://rewardingplay.com/2012/01/09/design-by-numbers-cooldowns/>
- [120] D. King, “Principals of UI design in the World of Warcraft,” Dec. 2019. [Online]. Available: <https://medium.com/@d.w.king12/principals-of-ui-design-in-the-world-of-warcraft-19e1a33feb61>
- [121] N. Babich, “The UI/UX design of progress indicator [Trends Examples],” May 2019. [Online]. Available: <https://usersnap.com/blog/progress-indicators/>
- [122] T. Jakobsen, “Advanced character physics,” in *Proc. Game Developers Conf.*, 2001, pp. 1–19.
- [123] D. Kao and D. F. Harrell, “Exploring the impact of avatar color on game experience in educational games,” in *Proc. CHI Conf. Extended Abstr. Human Factors Comput. Syst.*, 2016, pp. 1896–1905.
- [124] J. Stephenson, “A UX analysis of first-person shooter damage indicators,” Mar. 2018. [Online]. Available: <https://medium.com/@jasper.stephenson/a-ux-analysis-of-first-person-shooter-damage-indicators-59ac9d41caf8>
- [125] D. Daniels, “Why some games feel better than others - Part 3,” Accessed: Apr. 22, 2021. [Online]. Available: <http://lowfierce.blogspot.com/2007/03/why-some-games-feel-better-than-others.html>

- [126] S. Hurricane, "Impact freeze," Accessed: Apr. 22, 2021. [Online]. Available: <https://sonichurricane.com/?p=1043>
- [127] N. Berbee, "Game feel: Why your death animation sucks," San Francisco, CA, USA, 2015. [Online]. Available: <https://www.gdcvault.com/play/1022759/Game-Feel-Why-Your-Death>
- [128] K. Franinović, K. Franinovic, and S. Serafin, *Sonic Interaction Design*. Cambridge, MA, USA: MIT Press, 2013.
- [129] L. E. Nacke, M. Grimshaw, L. E. Nacke, and M. Grimshaw, "Player-game interaction through affective sound," in *Game Sound Technology and Player Interaction: Concepts and Developments*. Hershey, PA, USA: IGI Global, Jan. 2011. [Online]. Available: <https://www.igi-global.com/gateway/chapter/46796>
- [130] M. Orozco, J. Silva, A. E. Saddik, and E. Petriu, "The role of haptics in games," in *Haptics Rendering and Applications*. London, U.K.: IntechOpen, Jan. 2012, pp. 217–234.
- [131] M. Sabbagh, "The art of designing visceral and engaging bullet time gunplay," Oct. 2015. [Online]. Available: <https://michelsabbagh.wordpress.com/2015/10/07/the-art-of-designing-visceral-and-engaging-bullet-time-gunplay/>
- [132] W. Porter, "A videogame history of bullet-time," 2010. [Online]. Available: <https://www.gamesradar.com/a-videogame-history-of-bullet-time/>
- [133] C. Nutt, "The magic of towerFall: Depth, simplicity, community," 2015. [Online]. Available: https://www.gamasutra.com/view/news/241970/The_magic_of_TowerFall_Depth_simplicity_community.php
- [134] K. Birdwell, "The Cabal: Valve's design process for creating half-life," 1999. [Online]. Available: https://www.gamasutra.com/view/feature/131815/the_cabal_valves_design_process_.php
- [135] F. Thomas and O. Johnston, *The Illusion of Life: Disney Animation*. New York, NY, USA: Abbeville Press, 1981.
- [136] C. Karunamuni, N. de Vries, and M. Alonso, "Designing fluid interfaces - WWDC 2018 - Videos," 2018. [Online]. Available: <https://developer.apple.com/videos/play/wwdc2018/803/>
- [137] N. Gitter, "Building fluid interfaces," Aug. 2018. [Online]. Available: <https://medium.com/@nathangitter/building-fluid-interfaces-ios-swift-9732bb934bf5>
- [138] B.-W. Chang and D. Ungar, "Animation: From cartoons to the user interface," in *Proc. 6th Annu. ACM Symp. User Interface Softw. Technol.*, 1993, pp. 45–55. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=168642.168647>
- [139] H. Alexander, "The quiet importance of idle animations," 2019. [Online]. Available: <https://kotaku.com/the-quiet-importance-of-idle-animations-1834564079>
- [140] J. Couture, "What makes a great idle animation? Devs share their favorites," 2018. [Online]. Available: https://www.gamasutra.com/view/news/318163/What_makes_a_great_idle_animation_Devs_share_their_favorites.php
- [141] I. Keren, "Scroll back: The theory and practice of cameras in side-scrollers," 2015. [Online]. Available: <https://www.gdcvault.com/play/1022243/Scroll-Back-The-Theory-and>
- [142] I. Keren, "Gamasutra: Itay Keren's blog - Scroll back: The theory and practice of cameras in side-scrollers," 2015. [Online]. Available: https://gamasutra.com/blogs/ItayKeren/20150511/243083/Scroll_Back_The_Theory_and_Practice_of_Cameras_in_SideScrollers.php
- [143] R. Meyer, *ITSP Camera Explained*, May 2013. [Online]. Available: https://www.youtube.com/watch?v=aAKwZt3aXQM&feature=emb_title&ab_channel=RyanMeyer
- [144] M. Christie, P. Olivier, and J.-M. Normand, "Camera control in computer graphics," *Comput. Graph. Forum*, vol. 27, no. 8, pp. 2197–2218, Dec. 2008.
- [145] M. Haigh-Hutchinson, "Fundamentals of real-time camera design," GDC'05 Talk, Accessed: Apr. 22, 2021. [Online]. Available: <https://www.gdcvault.com/play/1019980/Fundamentals-of-Real-Time-Camera>
- [146] M. Haigh-Hutchinson, *Real time cameras: A guide for game designers and developers*, 1st ed. San Francisco, CA, USA: CRC Press, Apr. 2009.
- [147] M. Haigh-Hutchinson, "Real-Time cameras - Navigation and occlusion," 2009. [Online]. Available: https://www.gamasutra.com/view/feature/132456/realtime_cameras_navigation_and_.php
- [148] P. Burelli, "Virtual cinematography in games: Investigating the impact on player experience," in *Proc. Int. Conf. Found. Digit. Games*, May 2013, pp. 1–8.
- [149] G. N. Yannakakis, H. P. Martínez, and A. Jhala, "Towards affective camera control in games," *User Model. User-Adapted Interact.*, vol. 20, no. 4, pp. 313–340, Oct. 2010.
- [150] D. Norman, *The Design of Everyday Things*. New York, NY, USA: Basic Books, 1988.
- [151] Nintendo, "Super Mario Bros. 3," Game [Nintendo Entertainment System (NES)], 1988.
- [152] Team Meat, "Super Meat Boy," 2010. [Online]. Available: <http://www.supermeatboy.com>
- [153] K. Hamaluk, "Super Mario World Physics," Jul. 2012. [Online]. Available: <http://blog.hamaluk.ca/posts/super-mario-world-physics/>
- [154] M. Thorson, *Level Design Workshop: Designing Celeste*, 2017. [Online]. Available: <https://www.youtube.com/watch?v=4RlpMhBKNr0&feature=youtu.be>
- [155] Matt Makes Games, "Celeste," 2018.
- [156] N. Berry, "Celeste player controller source code," 2018. [Online]. Available: <https://github.com/NoelFB/Celeste>
- [157] G. Fiedler, "Integration basics," Jun. 2004. [Online]. Available: http://127.0.0.1:1313/post/integration_basics/
- [158] A. Saltsman, "Canabalt," 2009.
- [159] Psyonix, "Rocket League," 2015.
- [160] Nintendo, "Super Mario Galaxy," 2007.
- [161] K. Calis, J. W. Nijman, T. Vellmann, and A. Drucker, "Disc Room," 2020.
- [162] Nintendo, "Super Smash Bros. Ultimate," 2018.
- [163] Nintendo, "The Legend of Zelda," 1986. [Online]. Available: https://en.wikipedia.org/w/index.php?title=The_Legend_of_Zelda&oldid=949668668
- [164] Nintendo, "Super Mario Bros. 2," Game [Nintendo Entertainment System (NES)], 1988.
- [165] Number None, "Braid," 2008.
- [166] Rovio Entertainment, "Angry Birds," Game [iOS], 2009. [Online]. Available: www.angrybirds.com
- [167] EA Sports BIG, "SSX Tricky," Game [PlayStation 2], 2001.
- [168] Irem Corporation, "R-Type," Game [Arcade], 1987.
- [169] M. Alldridge, "'R-Type' - Irem. 1987," Mar. 2014. [Online]. Available: <http://www.markalldridge.co.uk/r-type.html#>
- [170] From Software, "Dark Souls," Game [PlayStation 3], 2009.
- [171] Epic Games, "Gears of War," Game [Xbox 360], 2006.
- [172] Codemasters, "DiRT 3," 2011.
- [173] Ready At Dawn Studios, "God of War," 2018.
- [174] Sucker Punch Productions, "inFAMOUS: Second Son," 2014.
- [175] Flashbang Studios, "Off-road Velociraptor Safari," 2008.
- [176] Diction, "Velociraptor Massacre," Apr. 2012. [Online]. Available: https://www.youtube.com/watch?v=uH7URFM8rcx&feature=youtu.be&ab_channel=Diction
- [177] Teknopants, "Samurai Gunn," 2013.
- [178] J. Deighan, "Sound design for video games: A primer," Aug. 2019. [Online]. Available: https://www.gamasutra.com/blogs/JamesDeighan/20190823/349296/Sound_Design_for_Video_Games_A_Primer.php
- [179] A. Marks, *The Complete Guide to Game Audio: For Composers, Musicians, Sound Designers, and Game Developers*, 2nd ed. Burlington, MA, USA: Focal Press, 2009.
- [180] J. P. Zagal and M. Mateas, "Time in video games: A survey and analysis," *Simul. Gaming*, vol. 41, no. 6, pp. 844–868, Dec. 2010.
- [181] Taito Corporation, "Bubble Bobble," Game [Arcade], 1986.
- [182] QCF Design, "Drawkanoid," 2018. [Online]. Available: <http://www.drawkanoid.com/>
- [183] grapefrukt games, "Holedown," 2018. [Online]. Available: <https://holedown.com/>
- [184] Firaxis Games, "XCOM: Enemy Unknown," 2012.
- [185] Bethesda Softworks LLC, "Fallout 3," Game [PlayStation 3], 2008.
- [186] PopCap Games, "Bejeweled," 2001. [Online]. Available: <https://ea.com/en-gb/games/bejeweled>
- [187] G. Fiedler, "Fix your timestep!" Jun. 2004. [Online]. Available: http://127.0.0.1:1313/post/fix_your_timestep/
- [188] I. Iacovides, A. Cox, R. Kennedy, P. Cairns, and C. Jennett, "Removing the HUD: The impact of non-diegetic game elements and expertise on player involvement," in *Proc. Annu. Symp. Comput.-Human Interact. Play*, 2015, pp. 13–22.
- [189] Taito Corporation, "Rastan Saga," 1987.
- [190] S. Eiserloh, "Math for game programmers: Juicing your cameras with math," 2016. [Online]. Available: <https://www.gdcvault.com/play/1023557/Math-for-Game-Programmers-Juicing>
- [191] Shadow Planet Productions, "Insanely twisted shadow planet," 2011.
- [192] N. Burtnyk and M. Wein, "Computer-generated key-frame animation," *J. SMPTE*, vol. 80, no. 3, pp. 149–153, Mar. 1971.
- [193] J. Lasseter, "Principles of traditional animation applied to 3D computer animation," *Comput. Graph.*, vol. 21, no. 4, pp. 35–44, Jul. 1987.

- [194] M. Pichlmair and F. Kayali, "Intentions, expectations and the player," 2008. [Online]. Available: https://www.academia.edu/4989138/Intentions_Expectations_and_the_Player
- [195] K. Pulver, "Platforming ledge forgiveness," 2013. [Online]. Available: http://kpulv.com/123/Platforming_Ledge_Forgiveness/
- [196] M. Thorson, "A short thread on a few Celeste game-feel things," 2020. [Online]. Available: <https://twitter.com/MattThorson/status/1238338574220546049>
- [197] R. Bernhaupt, W. Ijsselsteijn, F. F. Mueller, M. Tscheligi, and D. Wixon, "Evaluating user experiences in games," in *Proc. 26th Annu. CHI Conf. Extended Abstr. Human Factors Comput. Syst.*, 2008, Art. no. 3905. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1358628.1358953>
- [198] K. Isbister and N. Schaffer, *Game Usability: Advice from the Experts for Advancing the Player Experience*. San Francisco, CA, USA: Morgan Kaufmann, 2008.
- [199] T. Westin, I. Hamilton, M. Hinn, and R. van Tol, "Building a Manifesto for game accessibility," San Francisco, CA, USA, 2015. [Online]. Available: <https://www.gdcvault.com/play/1021849/Building-a-Manifesto-for-Game>
- [200] "Game accessibility guidelines|A straightforward reference for inclusive game design." [Online]. Available: <http://gameaccessibilityguidelines.com/>
- [201] M. Johansen, M. Pichlmair, and S. Risi, "Squeezee—A tool for designing juicy effects," in *Proc. Extended Abstr. 2020 Annu. Symp. Comput.-Human Interact. Play*, 2020, pp. 282–286.
- [202] T. D. Pettersson, "SFXR," 2007. [Online]. Available: http://www.drpetter.se/project_sfxr.html
- [203] A. Nealen, A. Isaksen, and D. Gopstein, "Exploring game space using survival analysis," in *Proc. Int. Conf. Found. Digit. Games*, 2015, pp. 1–9.
- [204] A. Isaksen, D. Gopstein, J. Togelius, and A. Nealen, "Exploring game space of minimal action games via parameter tuning and survival analysis," *IEEE Trans. Games*, vol. 10, no. 2, pp. 182–194, Jun. 2018.
- [205] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: A mixed-initiative level design tool," in *Proc. 5th Int. Conf. Found. Digit. Games*, 2010, pp. 209–216.
- [206] M. Guzodial, N. Liao, and M. Riedl, "Co-creative level design via machine learning," Joint Proc. AIIDE Workshops co-located 14th AAAI Conf. Art. Intell. Interactive Digit. Entertainment, Nov. 2018. [Online]. Available: http://ceur-ws.org/Vol-2282/EXAG_126.pdf
- [207] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proc. Int. Conf. Found. Digit. Games*, 2013, pp. 1–8.
- [208] M. Cook, S. Colton, and J. Gow, "The ANGELINA videogame design system," *IEEE Trans. Comput. Intell. AI Games*, vol. 9, no. 2, pp. 192–203, Jun. 2017.
- [209] M. Cook, S. Colton, and J. Gow, "The ANGELINA videogame design system," *IEEE Trans. Comput. Intell. AI Games*, vol. 9, no. 3, pp. 254–266, Sep. 2017.
- [210] J. Levine *et al.*, "General video game playing," in *Artificial and Computational Intelligence in Games (Dagstuhl Follow-Ups)*, vol. 6, S. M. M. Lucas, M. Mateas, P. Preuss Spronck, and J. Togelius, Eds. Wadern, Germany: Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, 2013, pp. 77–83.
- [211] A. Khalifa, M. C. Green, D. Perez-Liebana, and J. Togelius, "General video game rule generation," in *Proc. IEEE Conf. Comput. Intell. Games*, 2017, pp. 170–177.
- [212] R. D. Gaina *et al.*, "The 2016 two-player GVGAI competition," *IEEE Trans. Games*, vol. 10, no. 2, pp. 209–220, Jun. 2018.
- [213] D. Perez-Liebana, J. Liu, A. Khalifa, R. D. Gaina, J. Togelius, and S. M. Lucas, "General video game AI: A multi-track framework for evaluating agents, games, and content generation algorithms," *IEEE Trans. Games*, vol. 11, no. 3, pp. 195–214, Sep. 2019.
- [214] M. Johansen, M. Pichlmair, and S. Risi, "Video game description language environment for unity machine learning agents," in *Proc. IEEE Conf. Games*, 2019, pp. 1–8.
- [215] T. Schaul, "A video game description language for model-based or interactive learning," in *Proc. IEEE Conf. Comput. Intell. Games*, 2013, pp. 1–8.
- [216] B. Keogh, "An incomplete game feel reader," Mar. 2017. [Online]. Available: <https://brkeogh.com/2017/03/31/an-incomplete-game-feel-reader/>



Martin Pichlmair received the Dr. Techn. degree in computer science from the Vienna University of Technology, Vienna, Austria, in 2007.

He is currently an Associate Professor and the Head of the Games Programme with the IT University of Copenhagen, Copenhagen, Denmark, and a co-founder of the Vienna-based indie games studio Broken Rules. His current research interests include procedural and AI-assisted tools for game making.



Mads Johansen received the M.Sc. degree in games from the IT University of Copenhagen, Copenhagen, Denmark, in 2014.

He is currently a Ph.D. Fellow with the IT University of Copenhagen and a Co-Founder of the former Copenhagen-based game studio Glitchnap. His research interest includes mixed-initiative game development tools, with a focus on the prototyping phase of game development.