Combat in Games

Joseph C. Osborn, Dylan Lederle-Ensign, Noah Wardrip-Fruin, Michael Mateas
Center for Games and Playable Media, UC Santa Cruz
Santa Cruz, CA 95064, USA
{jcosborn,dlederle,nwf,michaelm}@soe.ucsc.edu

ABSTRACT

While the game design and game studies communities have analyzed combat both in specific games and game genres, and while combat is clearly central to many types of games, there is no general account of combat that is portable across diverse games. We provide such an account in the form of criteria which are satisfied by games that players interpret as "having combat." These requirements are eventually fulfilled via operational logics, which tie the game's observable behavior (including its instantial assets) to play experiences and cultural knowledge, creating what we refer to as a "combat model." In addition to establishing a comprehensive model of combat, making it possible to discuss combat across game genres, this work is the first to describe how complex playable models are constructed from *compositions* of operational logics working in concert; we also define two families of logics which are novel in the literature. This broad model of combat has already proved useful in practice, yielding insights in the analysis of the art game *Unmanned*; it also promises exciting computational applications in areas such as game design support tools and general game playing.

Categories and Subject Descriptors

K.8.0 [**Personal Computing**]: General—Games; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalism and Methods—*Representations (procedural and rule-based)*

General Terms

Game studies, game design

Keywords

Combat, operational logics, playable models

1. INTRODUCTION

Combat is widespread in games, but game studies has done little to investigate it as a broad phenomenon. Unlike game space [27, 23], game time [33, 15], or game fiction [11], there are no well known academic works dedicated to investigating combat across different sorts of games. Even the numerous essays and presentations on combat in the game design literature (e.g., [26, 6, 25, 3, 20, 1, 5]) generally work from informal or game-genre-specific conceptions of combat.

Doubtless one of the reasons combat is found in so many games is that its model, conventions for its use, and the tools for implementing it are remarkably well developed and easily accessible. This is, of course, no accident. The model, conventions, and tools were developed in the service of military training and planning, from Prussian *kriegsspiel* to the numerical war simulations of RAND and the visual flight simulations of Evans & Sutherland (from which today's numerical and graphical simulations of combat descend). Game studies texts such as Dyer-Witheford and de Peuter's discussion of Full Spectrum Warrior [9] have examined the historical evolution of game combat into the present day.

Game designers are quite skilled at modeling combat in specific game genres. Game studies analyses of combat tend to explore mainly its cultural context [14, 22]. Both kinds of knowledge are generally tied to narrow families of games: craft knowledge of fighting game design barely transfers to RPG combat, while the thematic gap between the military grand strategy game and the fantasy MMO would make a game studies analysis difficult to apply across those contexts.

In this work we develop a general account of combat, describing the specific interpretive moves required for a player to read a game as "having combat." These modeling obligations capture what is essential and what varies from game to game. We will limit our use of the word "combat" to exclude cases where actors merely have conflicting goals and mainly interact via those goals. Consider a competitive game like *Ticket to Ride,* in which players vie to claim lucrative cross-country railroad routes and may not share or trade the individual city-to-city linkages that comprise them. While the players are in direct conflict, they can only interfere with each other by claiming links before their opponents; they cannot for example steal other players' established routes or sabotage their trains. The players are in conflict, but it would be difficult to argue that combat is taking place.

We also specifically separate combat from contextualized battles where one or both sides aim to besiege, escort, sabotage, capture, defend, or destroy some inanimate object or noncombatant. These situations often include combat, but they are different systems—i.e., they have different interpretive obligations—from combat. For example, the passing and stealing elements of basketball could perhaps be said

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
Proceedings of the 10th International Conference on the Foundations of Digital Games (FDG 2015), June 22-25, 2015, Pacific Grove, CA, USA.
ISBN 978-0-9913982-4-9. Copyright held by author(s).

to model combat, but actually shooting for the basket cannot be accounted for as a combat maneuver. Wolf makes a similar distinction between symmetric and asymmetric conflicts, as in "combat" versus "shoot-em-up" versus "target" games [32].

Another goal of this paper is to contribute to the larger project of developing the concepts of operational logics and playable models, which form the theoretical foundation for our work [30, 31, 21, 28]. Operational logics are combinations of abstract processes with their communicative roles in a game, connected through an ongoing game state presentation and supporting a gameplay experience. Commonly-cited examples range from collision detection to resource sinks. Operational logics are critical building blocks for the construction of playable models, the underlying simulations that underwrite the space of player action. Playable models, which support players in incrementally exploring and learning actionable models for forming gameplay intentions, are used in games to model everything from physical space to economic systems, social relations, character development, and combat. We are not arguing that operational logics and playable models are the only useful lenses for thinking about games or computational media generally; rather, they provide a powerful analytical tool for connecting process-oriented and interpretation-oriented views of games.

This paper, through developing a model of combat, provides the first account of a complex playable model built up from heterogeneous operational logics. Understanding the composition of logics into a playable model supports analysis, interpretation and development by providing a robust alternative to the somewhat incoherent "genre" categories within which such activities typically take place.

This project was made possible in part by the Institute of Museum and Library Services (grant number LG-06-13-020-13).

2. OPERATIONAL LOGICS

Operational logics provide a way of thinking and talking about the connections between games as systems and games as media. The necessity of understanding such connections is emphasized in many of the most influential writings about games. Jesper Juul's 2005 book *Half-Real,* for example, foregrounds the connections between a game's rules and its fiction [17]. Similarly, the "Mechanics, Dynamics, Aesthetics" framework, which emphasizes the connections between systems and player experiences, has for years been at the core of the Game Design Workshop offered at the Game Developers Conference [13].

How can we follow through on what these ideas recommend: that we think about how games function and how they communicate at the same time? While other frameworks give names to the different categories one must think about (e.g., Juul's "rules" and "fiction") they offer no guidance for how to conceptualize or discuss entities that cut across the categories. This is also true of work that focuses specifically on conceptualizing game entities. For example, Raph Koster's "A Grammar of Gameplay" entirely brackets how games are experienced as media [18]. Koster's work is in dialogue with that of Ben Cousins, who writes about the "primary elements" of games as the conscious player interactions that cannot be further subdivided [8]. In practice, this put Cousins's work at the level of game mechanics (e.g., "jump" or "shoot") with no way to talk about the logics that support them.

The approaches of semiotics — from a variety of traditions — seem likely to help address this in the future. And the influence of semiotic thinking on the concepts of logics and models, and work that has been done with them, is undeniable. But for now even the most developed semiotic approaches tend to bracket the actual systems of games, instead focusing on game processes (to the extent they do) only as experienced by the player. For example, William Huber's 2012 dissertation, building on the tradition of Peircean semiotics, makes the argument that "the basis of the player's engagement with the digital game is the interpretation of a stream of signs" [12]. Whether we accept this or not, such a framing leaves no space for talking about the operations of the game system. Similarly, approaches that adopt a "pattern language" approach to thinking about games also tend to bracket consideration of processes, mainly focusing on player experiences [19, 4].

Operational logics and playable models address this lack. On a detailed level, looking at an operational logic (such as collision detection) both names a general strategy (how it combines an abstract process and a communicative goal) and gives a way of talking about how a particular game, or part of a game, employs the strategy (the specific algorithmic implementation, game state representation, and player experience). For example, at the level of operational logics, we interpret moving patterns of pixels—sprites—that stop when touching each other to be "objects" which "collide." Sudden upwards vertical movement of such a sprite followed by its rapid descent is read as "jumping" in the presence of "gravity" in a graphical logic. When our teletype prints out a block of text explaining "You are in. . . ", our sense of place among linked rooms requires that this text does not describe a different room when we "look" again, but does when we "go up stairs" or "enter portal" via a linking logic. Operational logics provide the combinations of media communication and system behaviors necessary to make sense of the screen.

What about more complex systems like combat, cooking, or politics, which may combine resource logics, spatial logics, and others? How do players read a game as representative of such a higher-order system? Certainly, cultural knowledge plays a large role, as recognized in the approach of proceduralist readings [29]: A set of game rules and assets describes cooking if "ingredients" in physical proximity are "prepared" (often using "tools") and combined into a food "product." Knowledge of common ingredients, cooking techniques, and foodstuffs can be attached to in-game phenomena such as sprites and game-mechanical reenactments of activities such as shaking a pan.

This assignment of game concepts to cultural concepts—ingredients, preparation processes, tools, products—is mandatory interpretive legwork that must always appear when the cultural frame of cooking is invoked. The concrete game concepts must meet certain requirements but are allowed to vary considerably: for example, ingredients must change form during preparation and be destroyed during combination, but ingredients and products could be resources of a resource logic or objects of a graphical logic. Their spatial collocation could mean proximity in a 2D spatial logic, their presence in the same inventory bag, or the accumulation of enough resource units corresponding to each ingredient.

Meeting these requirements requires additional interpre-

tive steps, grounded by the respective operational logics deployed to satisfy them. The combination of ingredients might be a conversion of two units of flour for one of bread in a resource logic, or the graphical collision of a bun and a patty after which both vanish and a complete hamburger appears. If the ingredients are still present in their original form, or if no product results, cooking has not taken place—we can only read those behaviors as bugs or as the mere *theme* of cooking divorced from its essential procedures. Any claim that a game models cooking is incomplete if it fails to define the ingredients, tools, preparations, and products in terms of the game's lower level logics in a way that is consistent with some cultural knowledge of cooking. Thus, at a broader level of analysis, playable models compose structuring information with varying operational logics to support the player in incremental exploration, intention formation, and interpretation.

Previous work on operational logics has focused primarily on two families of logics: graphical and resource logics. While these are used in many games' models of combat, there are additional logics at work that have not been explicitly identified. One such low-level logic is the *state logic*, which includes state machines of various stripes: finite state machines, extended finite state machines, state machines with timed transitions, and so on. These can be attached to individual game entities (in which case they are often communicated via sprite animations or decorations on the character's visual representation) or to the game as a whole (usually expressed via menus, UI elements, or general game screen composition). Considering fighting games specifically, *pattern matching* logics also play a critical role. Just as Tetris or Bejeweled feature *spatial pattern matching logics* (making patterns such as lines or groups), special moves and combos in Street Fighter or playing a successful series of notes in Guitar Hero are phenomena best understood as functions of a *temporal pattern matching logic.*

3. A MODEL OF COMBAT

We now explicate a class of playable models accounting for "combat" across a wide range of games and genres. Combat is integral to many games, where it serves as a central organizing activity. The fighting game, role playing game, first person shooter, and 3D action game (among others) all center on combat as the primary test of skill. Even puzzle games such as *Puzzle Quest* or *Puzzle Fighter* often frame the player's activity with the metaphor of a duel against an opponent. The language of combat (attacks and counterattacks) extends even more broadly to describe elements of competitive play in Chess and basketball. Which of these rhetorical uses of "combat" can really be said to model combat? In other words, what interpretive moves must be made—and, particularly, what phenomena and logics must be present—to claim that a game models combat?

We aim to characterize how players recognize a game as representing combat, regardless of the specific game or even genre. To this end we will examine two cases which differ in nearly every game-mechanical respect, but which nonetheless both model combat: the martial arts duels of *Street Fighter 2* and the medieval fantasy battles of *Final Fantasy.* It is clear that both games include combat as a central activity, but they have very little in common in terms of observed phenomena and underlying simulation: One is real-time and the other is turn-taking; one takes place in continuous space and the other in an abstract, trivial space of here-versus-there; one is a duel and the other involves opposing teams. The only game-mechanical similarity is that agents in each game's combat have health which is depleted by attacks, but we nonetheless recognize both situations as combat. From these two games we will extract a set of conditions which are sufficient and necessary to justify the claim that a game models combat. Finally, we will interpret *Quake* through the lens of these requirements to help validate our meta-model.

The key advantage of operational logics as an analytical tool is that they can connect a player's observations of a game's behavior to the underlying processes at work from the author's perspective, accounting for concrete instantial assets, abstract behaviors, *and* the linkages between them. Our approach will be to first enumerate, for each of our examples, each phenomenon that seems related to combat in its common-language sense, identifying the logic or logics that enact that relationship. We will then examine the roles fulfilled by the various logics at work in both examples and see what interpretive steps must be performed for models of combat in general.

This approach may seem circular, but we are explicitly aiming for a *descriptive* model usable in argumentation: capturing what seems to vary in the representation of combat across two very different games, and then using that abstraction to describe combat in other games. While we cannot hope for an exhaustive treatment given space limitations, we have tried to achieve some generality by selecting games which have little in common mechanically. Concretely, we are answering the question: what interpretive steps are required to read a game's elements as modeling combat?

3.1 Street Fighter 2

Fighting games focus nearly all their systems and instantial assets on modeling combat, specifically the martial (and supernatural) arts duel popularized by *Street Fighter 2.* We ignore the best-of-three match structure and the single player mode here for simplicity.

Our first objective is to explain what the players see immediately when starting a match. First and foremost are the large and luxuriously animated fighters which move in continuous 2D space with the corresponding player's joystick (strong graphical-logic evidence for control and agency). These burly martial artists square off against each other from a respectful distance; their faces match the portraits seen at the character selection screen, and their poses make it clear that they intend to do battle. Light, size, and color contrast these fighters, and the ground on which they walk, against a background with which players cannot interact. The theory of operational logics has not yet committed to whether these cues of visual communication are associated with specific graphical logics or only with the representation strategies of other logics; in other words, whether they are logics of their own or are only authoring conventions.

A player may notice that walking away from her opponent does not change the direction her own fighter is facing. This cue implies that the two are enemies and reinforces the interpretation that we are enacting a martial arts duel. Characters are given additional weight, and their adversarial relationship is further emphasized, by the fact that they cannot walk through each other (a conclusion drawn from a collision logic). The fighters may jump, and indeed may jump over each other—turning around to maintain eye con-

tact in the process—which suggests a world where physics logics control the movement of objects on the screen.

At the top of the screen, there are two yellow bars with each fighter's name underneath: player one's bar on the left where player one's fighter stands, and player two's on the right. Between the two bars sit the letters 'KO' and a timer counting down. In the context of a martial arts fight, this plainly means "knockout" and the timer is a limit on the length of the match—we know that the fight will end either with one fighter knocked out or because the time runs out. This reading implies that the match functions as a tiny state machine, starting in a combat state and transitioning to victory for one or the other player or else a draw.

Experimenting with the six buttons next to each joystick sends that player's fighter into a flurry of animations which look like punches and kicks—the upper buttons punch and the lower buttons kick, connecting the physical controls to the characters' bodies. Different fighters have different punches and kicks as appropriate for their size and martial arts school. If the attacker is close enough to their opponent during such an animation, the latter plays an animation superficially indicating pain in response, and one of the yellow bars shrinks, revealing a red bar. We can therefore interpret this yellow bar as indicating a current value out of a maximum defined by the red bar: a graphical representation of a resource logic.

The KO glyph is in between the two bars, and the yellow bars shrink towards it; these graphical hints suggest that, if a yellow bar empties, that fighter is out cold and the match is over. We can also imagine that whoever has the longer yellow bar when time runs out wins by default, which is consistent with our understanding of technical wins in formal martial arts. Both of these are confirmed through play. This ties the transitions of the match's state logic to the resource logic of the yellow bars, each of which we can call "health" or "stamina" by its relation to being knocked out.

The transactions of the resource logic are themselves triggered by the collision logics associated with punches and kicks and, specifically, the reactions of the attacked characters. If this latter connection were absent—if punches and kicks did nothing, or if health decreased for no visible reason when buttons were pressed—it would be difficult to argue that the martial artists were fighting.

Further reinforcing the fiction of a fighter being injured, any attacks which are "in flight" when a combatant is hit are canceled, and the player's buttons and joystick are nonresponsive for a brief interval following each hit; in fact, characters also ignore joystick inputs while attacking. Something similar happens when two attacks of comparable strength connect with each other: both fighters are hurt, or else both attacks are neutralized without damaging either party. "Hit-stunned" and "attacking" are time-bounded states operating in per-character state machine logics, constraining the available inputs and showing special animations to give impressions of kinesthetics, bodily weight, and even attack strength. Attacks which seem stronger (e.g., involve more wind-up or follow-through in the animation) *feel* stronger, and accordingly do more damage.

If a fighter is backing away from their opponent during an attack, they put up their arms to block (entering a blocking state). We know this functions as a block because when hit their health bar decreases less, the hit stun is shorter, and the defender is much less likely to be knocked flat out (resource and state logics at work). The counter to blocking is throwing, activated by pressing a punch and kick button simultaneously (a "whole body" move). Blocking will not protect against throws, which is consistent with our cultural knowledge that throws involve grappling, not percussive force; throws also drop the thrown character to the ground.

Jumping and crouching (holding down on the joystick) can both be used to dodge attacks by their involvement with the collision logics. Perhaps more importantly, they act in the character state logic to change the function of every attack button. This is consistent with a notion of combat stances borrowed from martial arts, and gives more predictable access to the three heights at which characters are vulnerable: the feet, the body, and the head.

Finally, the *Street Fighter* series in particular is known for character-specific special moves. If a curious or well-informed player inputs joystick directions and attack buttons in certain preordained, time-sensitive patterns (which vary from fighter to fighter), their fighter performs none of the individual actions entered, but rather plays a custom animation with an unusual combat behavior: flinging a fireball across the screen, flying across the stage like a torpedo, delivering a punishing spinning uppercut, shocking adjacent opponents with electricity. This deployment of a temporal pattern matching logic makes it clear that the fighters are world-class martial artists extensively trained in secret techniques that transcend mere brawling with fists and feet. Only players who are both in the know *and* meet a threshold of manual dexterity can perform these moves (intentionally), just as the characters themselves have inherited the collected knowledge of their respective schools.

To sum up, these are the ways that *Street Fighter* deploys operational logics to evoke the combat seen in (fictionalized) martial arts duels:

- graphical logics suggest two dueling fighters in an arena
- graphical, physics, and collision logics help us read movement as advancing, positioning, and dodging
- graphical, state, and temporal pattern matching logics communicate physical, kinesthetic actions with different attacks using the body in different ways
- graphical and state logics determine success of attacks
- graphical, state, and resource logics work together to express fighters' reactions to being hit
- state and resource logics structure matches

3.2 Final Fantasy

Whereas fighting games ground combat in real-time graphical and state logics, computer role-playing games (RPGs) model it mainly via resource logics, often with turn-taking. The early console RPG *Final Fantasy* cemented genre conventions of party combat that were inspired by earlier games including tabletop RPGs. Combat, exploration, narrative progression, and character enhancement comprise the four pillars of *Final Fantasy* and its successors, but here we restrict our attention to the battle scenes.

As in our reading of *Street Fighter,* we begin by explaining the initial state of the combat screen. On the right we see the sprites of the four heroes selected by the player at

the beginning of the game, arrayed vertically and all facing left; the first is slightly to the left of the others. They are on a black field surrounded by a white rectangle whose upper quarter has an environmental background appropriate to wherever the fight began (e.g., a forest or a dungeon). These cues suggest that the heroes are in the same space, and that it is connected to their position on the world map. Their rigid lineup amounts to a battle formation, and the first hero—stepping ahead of the rest—is in a distinguished position. The player might eventually intuit that the frontmost characters in the party are more likely to be targeted by enemy attacks; this connects the graphical logic to the resource logics of combat.

The screen has several other boxes, and by now we can recognize them as something like overlapping windows: some of them occlude the edges of others. Each of the windows to the right of the heroes contains the name of a character, the glyph "HP," and a number. These are laid out in the same vertical order as the heroes, suggesting (along with the names) that each is bound somehow to the corresponding character.

The largest window is on the left and contains sprites that vary from fight to fight. They are larger than the heroes and more intricately illustrated, always face right, and look like frightening monsters. These are all cues implying threat and conflict. This box also has the environmental features used in the heroes' window. This tells us that the monsters are in the same general area as the heroes, but the window boundaries and width of the black fields suggest a substantial distance between the groups.

Below the monsters is a box with one or more words. Some experimentation shows that these vary with the monsters encountered and vanish when all the monsters with the same sprite are defeated, so over time the player understands them as the names of types of monsters.

The last window we see when starting a fight is a grid of fantasy-combat actions: "Fight," "[Cast] Magic," and so on. A cartoon finger points at "Fight," and moving the directional pad moves the finger from word to word. By analogy to graphical adventure games, we can guess that some model of abstract commands is active here. The position of the hand—a symbol of agency—is tied to a choice of actions which will be enacted by some other logic after selection. Instead of "LOOK" or "TAKE", our verbs are mainly violence against the enemies on the left.

Pressing the A button while "Fight" is selected moves the hand over to the enemies. The same logic that denoted selecting an action is now used to select a target. Pressing the B button moves the hand back, canceling the initial choice (a state logic at work). Pressing the A button also moves the hand back to the action list, but the first character moves backwards to the right and the second steps forward—but the "Fight" action has not yet taken place! This strongly suggests that we have merely committed to an action for the first hero and are now making a similar decision for the second hero. This impression is reinforced by the observation that pressing B now moves the second character back rightwards and the first left again. State and graphical logics combine to give form to the idea of selecting an action for each hero before resolving those actions, a convention shared with some pre-digital role playing games.

Selecting "Magic" causes another window to pop up and moves the hand there. This window overlaps the others, suggesting hierarchy: we are now choosing a specific spell such as "Fire" or "Cure." Not all heroes have the same choices here—the martial artist (recognized by his headband) knows no magic, and the spells of the white and black mage are disjoint. Individual spells are purchased and assigned to characters one by one, so even two black mages may have different options. These effectively form per-character magical inventories, a simple abstract spatial logic where each hero "has" certain spells contained within the abstract space of their memory.

Merely knowing a spell is not enough to select ("cast") it. Magic comes in tiered power levels; magic users can only cast spells of a given level a few times before they run out of "magic points" at that level and must replenish them by sleeping. These limitations on magic are given by a resource logic layered over the inventory of spells, surfaced by numbers at the left of each row (tier) of spells in the menu. Once a spell is chosen, it is directed—using the pointing hand—at an enemy (for single-target offensive spells), an ally (for single-target defensive or healing spells), or to the entire enemy or ally party (for group-targeted spells).

"Drink" gives a choice of healing potion from the party's shared inventory and uses it on the current character. This inventory has a set number of slots occupied by stackable and non-stackable items; stackable items may fit up to 99 to a slot (another spatial logic of containment with aggregation). "Item" permits the use of individual heroes' carried equipment as items in battle. Most have no effect but some function akin to spells without the magic point restriction. Each hero has four slots for weapons and four slots for armor (containment without aggregation). "Run" takes no target and indicates an intent to flee the battle on behalf of the entire party; each hero who tries to run gives the whole group a chance to escape.

After all the actions are selected, the combat resolution phase begins. Allies and enemies respectively (in a randomly determined order which models the uncertainty of combat) animate and flash to indicate acting and reacting as more windows pop up over the screen. The attack animations and hit reactions are a simpler version of what we see in *Street Fighter 2*. These message windows appear in consistent locations, with one indicating the initiator of an action, another the (current) target, and additional windows showing the effects. During the fight, the numbers in the right hand windows will decrease when the corresponding heroes are attacked, representing the resource logic of hit points (character health). It can be assumed that enemies have a similar resource, and when it is exhausted the enemy vanishes (and subsequent attacks against them are useless) or the hero collapses, unable to select or perform any actions. When an entire team is defeated (or the heroes successfully flee), the battle ends in victory or defeat (a tiny state logic).

An attack may miss, hit, or critically hit, determined by the relative statistics of the attacker and defender; we read those outcomes textually and they are consistent with observed changes to characters' hit points. Basic attacks (the "Fight" command) can hit multiple times based on the implied speed and the accuracy statistic of the attacker: the dexterous ninja and master martial artist strike the most times, the knight hits fewer times but for more damage per strike, and the frail mages attack the slowest (here, the resource logic connects to cultural knowledge of physical prowess). The resulting damage depends on these statistics

in yet different ways (and the specific attacks: for example, some spells hurt only undead monsters), and the statistics in turn depend on the equipment of the heroes involved. The systems of equations governing the resource logic of hit points are the central feature of RPG combat, and there are many branches, conditions, and modulations contingent on mostly-invisible numbers. The abstract and non-spatial modeling of strikes here and the substantial role of random numbers is a strong differentiator between combat in *Final Fantasy* and combat in *Street Fighter 2,* with the former being much less predictable and more chaotic.

Besides reducing hit points, a variety of other effects can occur in battle (mainly through spells or monster abilities). Hit points of living characters can be restored by healing spells or potions; a dead character can be revived by certain specialized spells. Characters can also be made stronger or faster; be prevented from acting by magical sleep, paralysis, or petrification; take gradual damage from poison; be prevented from casting spells by having their voices silenced; and gain resistances or immunities to specific elements. Some of these effects are resource augmentations and others are character states in a state logic. The multiplicity of modifiers that combatants can obtain is another distinguishing feature of RPG combat.

In summary, these are the phenomena of *Final Fantasy* battles that contribute to its model of combat:

- graphical logics show an enemy and ally party arrayed against each other in an abstracted environment
- graphical and state logics interoperate to describe the flow of combat in phases: selecting fantasy-themed actions and targets for each hero, and then resolving those actions in a loop
- resource, space (inventory), and state logics constrain the space of possible actions
- resource and state logics determine success or failure
- resource, state, and graphical logics calculate and render the effects of combat actions
- state and resource logics structure combat as a whole, determining the final outcome

3.3 An Abstract Model

Having examined combat in two wildly different games, we can see what interpretive steps appear across depictions of combat. First, both games feature multiple combatants (sometimes on teams) in a shared space: the arena or the battle screen. These actors aim to do violence to each other and may perform nonviolent actions like movements or beneficial spells to enable future violence or to prevent, mitigate, or undo violence committed upon them or their allies.

Violence here is broadly construed and is not limited to health damage: forcibly moving an opponent, hindering its ability to act, and stopping it from preventing, mitigating, or undoing violence are all violent acts because they are all non-consensual. Some actions could have both nonviolent and violent components (e.g., a parry which protects the defender and stuns the attacker, as opposed to a dodge), and some could be contextually violent (e.g., bumper cars only move around, but some movements impinge on other actors' personal autonomy).

While not all actions performed in combat are violent, every combat actor must aim to do violence or support the violence of its teammates or else it is a noncombatant. This immediately distinguishes combat from attacking inanimate objects or the defenseless: combat requires a degree of symmetry and a mutual agreement to fight. Any model of combat must support a claim that violence is being done by all involved sides. This exchange of violent (and sometimes nonviolent) actions is realized by the real-time graphical, state, and collision logics of *Street Fighter 2* and the turn-taking graphical, state, and resource logics of *Final Fantasy.*

We can pin down the exchange of combat-related actions a little more firmly: At a given time, combatants have certain actions available and may select one to activate, possibly with parameters such as one or more targets; this action conditionally succeeds; the degree of its success is likewise conditional; and the target or targets (and possibly the initiator as well) react in response to it. Each of these is a hole into which different models supported by different operational logics can be slotted, but every set of phenomena that players interpret as representing combat must suggest linkages—however trivial—satisfying each component.

Finally, combat must come to an end, either altogether or just for certain unlucky combatants. State logics are often deployed alongside resource logics of health or points for this purpose, but occasionally a fighter can be disqualified for leaving or being forced outside of the combat space (often via graphical logics). In general, violent actions will try to bring about a bad end for enemy actors and nonviolent actions will facilitate that violence, prevent that bad outcome, or produce a good outcome for allies. This accounts for battles whose aim is a rout as well as those fought for points.

To faithfully model combat in the sense used by this paper—a sense which accounts for substantial variation across games without being uselessly broad—a game must provide:

- 1) A space in which 2) multiple agents 3) exchange violent (and possibly nonviolent) actions with each other
- 4) A way to decide which actions an agent may perform, 5) whether they succeed, and 6) to what extent
- 7) Observable effects for all combat actions which are eventually visible at least to the initiator and target(s)
- 8) Circumstances under which agents enter or exit the fight and 9) for the combat itself to terminate

Games which admit interpretations satisfying these nine conditions will necessarily also model "conflict;" it is a weaker claim to say that a game models conflict versus combat. In the same way, "cooking" can be read as "crafting," but this is a weaker interpretation—"cooking" additionally requires that the products and ingredients seem like food, and that the processes evoke the tools and methods of cookery. Magically summoning food is not a substantial model of cooking, and neither is making a clay sculpture, even though the former produces food and the latter uses baking to convert resources of one type into another. So many games prominently feature conflict that it is often seen as a fundamental quality of games [16, 2], but we believe it is useful to separate agent-versus-agent conflict (as in combat) from e.g. player-versus-environment conflict.

We will first test our abstracted model on games that are similar to *Street Fighter 2* and *Final Fantasy* to see how it accounts for relatively small changes.

Divekick is a two button fighting game drawing from the competitive *Street Fighter* community. It deploys operational logics in similar ways to *Street Fighter 2* but reduces the grammar of possible actions significantly; moreover, the fight ends after one attack connects. In terms of the combat model, there is no more role for a resource logic in structuring the match unless we view health as a trivial binary resource. Draws are determined by means of a graphical logic: the player closest to a red line drawn on the screen is the winner when time runs out. While a *Street Fighter* character has movement, kicks, punches, blocks and extensive special moves, *Divekick* strips these verbs away—including directional movement—and replaces them with only two inputs used for both movement and attacking. Even with the resource and temporal pattern matching logics completely removed and character behaviors drastically simplified, the game still models combat. *Divekick* meets our requirements in different ways but still meets them without trouble.

Another tweak to the *Street Fighter 2* formula comes in the 3D arena fighter. Here, we consider *Soul Calibur 2.* Nearly every requirement for modeling combat is fulfilled in the same way by a similar logic, except that combat takes place in a 3D arena and both vulnerable and damage dealing regions are 3D volumes instead of sets of 2D boxes. The main consequence is that some attacks can be side-stepped by moving the vulnerable region in 3D space. There are other smaller changes: a well timed block can turn into a parry which stuns the attacker (we can interpret this parry as an action available only within a small time window), and being knocked out of the combat area is an instant loss. Switching to 3D logics and defining defeat with a spatial model does not break our interpretation of combat.

A primary way computer role-playing games differentiate themselves is by variations of their combat systems. Later *Final Fantasy* games would enhance their spatial models, deploying the concept of rows to modulate the success and degree of success of combat actions. Monsters could be too far away to hit with close range attacks, or front-row characters would deal and receive more damage from melee attacks.

In *Chrono Trigger,* actions are exchanged in real time: characters gradually accumulate a resource which, when full, permits them to select an action which is delivered a little while later depending on the action's speed. This extends the resource logics to also account for turn ordering. More interestingly, *Chrono Trigger* employs a relatively sophisticated model of space. Enemies are visible in the game's fields and dungeons, and bumping into them initiates combat. This does not take place on a new screen as in all our other examples, but the enemy and player parties spread out in formation on the same map, jostling and shuffling as combat proceeds. The relative positioning of characters does not matter with respect to standard attacks, but certain magic spells determine their success by testing for the presence of a target within a region, usually an oriented rectangle or a circle. The already more-coherent model of space is reinforced by these spatial effects, which incorporate graphical logics into the resource logic that forms the kernel of combat (around which the rest of the game revolves).

3.4 Quake

Combat is clearly the central activity of *Quake,* so *Quake* should satisfy the requirements we have defined (we focus here on the Deathmatch multiplayer mode). Much of this analysis extends to other shooters like *Unreal Tournament.*

The first requirement is a space to frame the fight. *Quake* play takes place in a continuous 3D space realized with 3D graphical and physics logics, which is perhaps its most distinctive feature compared to many fighting games. The fully playable model of space (and the level geometry that designers place within it) enables complex tactical combat: players have a wider degree of freedom to position themselves and launch combat actions than they do in the two dimensional play spaces of fighting games.

Combat actions in *Quake* are determined by the available weapon types (an inventory model like *Final Fantasy*'s items). Weapons are obtained via collision logics, as are the ammunition pickups they require (a resource logic at work). Players exchange actions using these weapons, each of which has a different health-reducing effect in the central resource logic. Health recovery and damage mitigation pickups are also obtained via collisions, as are special powerups that put the receiver into a powerful state (e.g., invisible or extra-strong). All these powerups are subject to state logics, reappearing some time after being picked up. Competitive play revolves around controlling access to these resources.

Our model of combat allows for non-violent actions that support combat, encompassing the dodging and positional jockeying that characterizes much of *Quake* deathmatch play. Switching weapons is another non-violent action which supports future combat by manipulating a per-character state logic.

Each violent action has its own way of determining success in the graphical and collision logics. Rockets and grenades have area effects in which "splash damage" is dealt, attenuating the effect with distance. On the other hand, a rail gun only deals damage at a single point in space. Every weapon engages with the resource logics of health and armor.

Quake actions have clear and observable effects for the player. Animations representing various types of explosions, as well as sound cues, indicate to the player success or failure of their actions, as well as the degree of their success. *Quake* also features a HUD, an interface that informs the player of their current state with regards to the resource logics of health, armor and ammo. This interface element supports combat actions (but does not execute them) and indicates the success of opponents' combat actions.

Players exit combat whenever they lose all their health, but they return after a few seconds; of course, they may also quit outright. Combat ends after a set timer or when all players quit the game. While some play activities like dodging or hiding may not be violent actions, they are performed only to gain or keep combat advantage. This contrasts with linear shooters like *Spec Ops* or adventure games like *Metroid Prime,* which engage the player with a series of short combat encounters interspersed with noncombat dialog, exploration, puzzle-solving, or cut scenes.

A major component of tactical play in *Quake* is aiming one's weapon, but aiming is not a violent action on its own. This poses no problem for our characterization of combat: while aiming is not violent, it clearly portends and supports future violence. In this sense, it is similar to movement within the level, or indeed to jockeying for position in *Street Fighter 2.* Positioning, aiming, and even shooting aren't necessarily tied to violence, though they certainly are in *Quake*: The same ballistic physics would come into play in a simulated game of darts, or in a game about extinguishing fires.

3.5 Unmanned

All of our examples thus far come from what we might call the "mainstream" games community. Unlike other common models in games (e.g., spatial models) combat models have rarely been used (or theorized) for art, political, documentary, queer, or other broader purposes. Celebrated indie games with combat models certainly exist (e.g., the fencing game *Nidhogg*) but rather than using combat models to explore something beyond combat, these generally offer some new experience of combat itself. In fact, indie games such as *Gone Home* and *Dear Esther* are often noted precisely for having removed combat from the types of implementations of real-time 3D spatial models with which it is so strongly associated by some game players.

This may seem puzzling. The game activity enabled by combat models—violence—is a subject commonly encountered in fine art and political critiques in other media, both in interpersonal forms and in broader conceptualizations such as "state violence." Using well-established combat models could enable games that address violence both literally and through metaphorical approaches (analogously to the use of spatial models in e.g. *Passage* and *Dys4ia*).

This might be due to politics inherent in combat. It is challenging to construct a game that employs the combat model without reproducing the assumption, for example, that violence is the correct approach to conflict. Consider the well-known tale of Elizabeth Magie's *The Landlord's Game,* which was intended as a critique but became a popular way to enact monopolist fantasies under its current title: *Monopoly* [10, 24]. On the other hand, independent games such as *Cart Life* suggest a way to subvert such dominant models. Playing *Cart Life* requires participating in capitalism and struggling to thrive in it, but the experience of playing the game explicates how the system is organized against powerless people like the player characters. It succeeds at a politics that critiques the game's own economic model, and the critique is not overwhelmed by that model.

Molleindustria and Jim Munroe's game *Unmanned* stands out for both employing and interrogating combat. The player character is a drone operator who lives in the western U.S. but remotely pilots military drones located in the Middle East. The game seems to be built on two models: a choice-based dialogue/story model and the combat model described in this paper. The dialogue/story model drives the experience of the operator talking to himself, his co-pilot, and his wife and son. The rest of the game consists of two representations of mediated violence: using the drone for tracking and shooting targeted people in the Middle East and playing a "contemporary warfare"-themed video game at home with the character's son.

But if we examine *Unmanned* using the combat model described here, it becomes immediately clear that only one of these activities actually employs combat *per se.* While the game-within-the-game employs it exactly, the drone operation portions have only one agent capable of taking violent action: the player character. This is therefore not a model of combat, but rather a model of something else—perhaps *hunting.* Only the state is capable of committing violence in this circumstance (this asymmetry is the norm for state violence). *Unmanned* uses a combat model in its game-within-the-game precisely to emphasize what most military-themed games elide: the model of combat in games is not the model of much of the violence carried out by the U.S. military—despite the rhetorical claims of games like *America's Army.*

4. APPLICATIONS

Beyond game studies, we believe operational logics can contribute to game-making tools and technical game research. Game-making tools often commit to specific combinations of operational logics: *GameMaker* to continuous-space 2D graphical logics, *Twine* to linking logics, *PuzzleScript* to spatial matching and collision logics, and so on.

Considering combat, the fighting game platform Mugen combines spatial, state, temporal pattern matching, and resource logics to model dueling in the style of *Street Fighter.* But Mugen provides only one of many possible models of combat. RPGMaker provides substantially different (and more constrained) authorial affordances for combat, minimizing the roles of spatial and matching logics. A formal understanding of combat in terms of operational logics—what is necessary to model it, how it fits into and uses a game's other logics—could yield a design language for combat which is concrete enough to implement in a game-making tool, and this remains important future work.

The explicit representation of combat or other playable models could also aid in the automated analysis of games. Michael Cook's *Mechanic Miner* struggled to interpret homogeneous source code variables and operations, having no *a priori* knowledge of what was being modeled [7]. Operational logics give a context for programs (as well as people) to interpret game rules. For example, if combat tends to lean on resource logics in a consistent way across games, a future *Mechanic Miner* could hypothesize the existence of a combat system and look for variables that fill the roles required of combat-related resources like health. A general game-playing AI could similarly try interpreting its game through the lenses of various operational logics and possible models to abstract the game under consideration and form high level strategies or transfer learned skills between games.

5. CONCLUSION

In this work, we have developed a comprehensive description of combat in games that spans diverse game genres and gameplay styles. Discussing combat as a concept separate from its concrete game-mechanical implementation enabled a novel analysis of the art game *Unmanned.* The tools we deployed to characterize combat in this way were built on proceduralist readings, which we informally extended to explicitly account for compositions of operational logics.

An obvious next step is to formalize and further operationalize the theory of operational logics and this model of combat. Specifically, understanding exactly *how* logics compose with each other to support sophisticated readings—not just *that* they compose—remains for future work. As a part of this process, it must be determined whether there are elements of a game such as cutscenes or rules of visual composition that cannot be adequately explained through operational logics, and if so to what extent operational logics should be expanded to cover them.

It is also important to apply this model to explain more instances of combat (and non-combat) in games (especially radically different games) to test its limits and validate its assumptions. Likewise, this line of reasoning can be used to determine the interpretive obligations of modeling other systems; a library of such models would be extremely useful.

6. REFERENCES

[1] M. Acero. Combat design 101: Creating cohesive combat systems. In *Game Design Expo Vancouver*, 2013.

[2] E. M. Avedon and B. Sutton-Smith. *The study of games*. Wiley New York et al., 1971.

[3] M. Birkhead. Depth vs breadth in combat design: An interactive visualization, May 2011.

[4] S. Bjork and J. Holopainen. *Patterns in game design (game development series)*. Charles River Media, 2004.

[5] A. Bormann. Controlling chaos: Designing compelling combat in action games. In *Game Connection*, 2014.

[6] J. Bridge. Anatomy of a combat zone. *Gamasutra*, 2009.

[7] M. Cook, S. Colton, A. Raad, and J. Gow. *Mechanic miner: Reflection-driven game mechanic discovery and level design*. Springer, 2013.

[8] B. Cousins. Elementary game design. *Develop magazine*, 2004.

[9] N. Dyer-Witheford and G. De Peuter. *Games of empire: Global capitalism and video games*, volume 29. U of Minnesota Press, 2009.

[10] M. Flanagan. *Critical play: radical game design*. MIT press, 2009.

[11] P. Harrigan and N. Wardrip-Fruin. *Second person: Role-playing and story in games and playable media*. The MIT Press, 2010.

[12] W. H. Huber. *The Foundations of Videogame Authorship*. PhD thesis, The University of California, San Diego, 2013.

[13] R. Hunicke, M. LeBlanc, and R. Zubek. Mda: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on Challenges in Game AI*, volume 4, 2004.

[14] A. Järvinen. Halo and the anatomy of the fps. *Game Studies*, 2(1):641–661, 2002.

[15] J. Juul. Introduction to game time. *First person: New media as story, performance, and game*, 1:131–141, 2004.

[16] J. Juul. The game, the player, the world: Looking for a heart of gameness. *PLURAIS-Revista Multidisciplinar da UNEB*, 1(2), 2010.

[17] J. Juul. *Half-real: Video games between real rules and fictional worlds*. MIT press, 2011.

[18] R. Koster. A grammar of gameplay. In *Game Developers Conference*, 2005.

[19] B. Kreimeier. The case for game design patterns. *Gamasutra*, 2002.

[20] S. Lambottin. The fundamental pillars of a combat system. *Gamasutra*, 2012.

[21] M. Mateas and N. Wardrip-Fruin. Defining operational logics. *Digital Games Research Association (DiGRA)*, 2009.

[22] B. W. Ng. Street fighter and the king of fighters in hong kong: A study of cultural consumption and localization of japanese games in an asian context. *Game Studies*, 6(1):2006, 2006.

[23] M. Nitsche. *Video game spaces: image, play, and structure in 3D game worlds*. MIT Press, 2008.

[24] M. Pilon. *The Monopolists: Obsession, Fury, and the Scandal Behind the World's Favorite Board Game*. Bloomsbury Publishing USA, 2015.

[25] B. Russell. A deeper look into the combat design of uncharted 2. *Gamasutra*, 2010.

[26] J. Song. Improving the combat 'impact' of action games. *Gamasutra*, 2005.

[27] L. Taylor. When seams fall apart: Video game space and the player. *Game Studies*, 3(2):26, 2003.

[28] M. Treanor, M. Mateas, and N. Wardrip-Fruin. Kaboom! is a many-splendored thing: An interpretation and design methodology for message-driven games using graphical logics. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, pages 224–231. ACM, 2010.

[29] M. Treanor, B. Schweizer, I. Bogost, and M. Mateas. Proceduralist readings: How to find meaning in games with graphical logics. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, pages 115–122. ACM, 2011.

[30] N. Wardrip-Fruin. Playable media and textual instruments. *Dichtung Digital*, 2005(1), 2005.

[31] N. Wardrip-Fruin. *Expressive Processing: On Process-Intensive Literature and Digital Media*. PhD thesis, Brown University, 2006.

[32] M. J. Wolf. *The medium of the video game*. University of Texas Press, 2001.

[33] J. P. Zagal and M. Mateas. Time in video games: A survey and analysis. *Simulation & Gaming*, 2010.