the international journal of computer game research
volume 8 issue 2
December 2008
ISSN:1604-7982

Miguel Sicart

Miguel Sicart received his PhD in game studies in December 2006. His 3-year research project focused on providing a multidisciplinary approach to ethics and computer games, focusing on issues on game design, violence and videogames, and the role of age-regulation codes. His research has now crystalized into the book, The Ethics of Computer Games (MIT Press, 2009). His current research focuses on developing a design framework for implementing ethical gameplay in digital games. Miguel Sicart is Associate Professor at the IT University of Copenhagen, where he teaches game design.

Email: miguel@itu.dk
www.miguelsicart.net

Defining Game Mechanics

by Miguel Sicart

Abstract

This article defins game mechanics in relation to rules and challenges. Game mechanics are methods invoked by agents for interacting with the game world. I apply this definition to a comparative analysis of the games *Rez*, *Every Extend Extra* and *Shadow of the Colossus* that will show the relevance of a formal definition of game mechanics.

Keywords Game Design, Game Research, Game Mechanics, Rules, Challenges.

Introduction

Gears of War (Epic Games, 2006) showcased the impressive graphical capacities of the then-called "Next Generation" consoles. Making good use of the XBox 360 hardware, *Gears of War* models, textures and general aesthetics excelled. Yet, it is likely that this game will be remembered not as an exhibition of what archaic technology could do, but as the title that popularized the cover mechanics in third-person action games. Inspired by the cover system of *kill.switch* (Namco, 2003), *Gears of War* combined a linear level structure with action sequences where the dominant strategy is to take cover and patiently create an effective combat tactic. The influence of this design choice is such that even titles like *Grand Theft Auto IV* (RockStar North, 2008) have implemented a cover mechanic. Taking cover has arguably become one the features that all triple-A third-person action games ought to have nowadays.

The question is: what does "mechanic" mean in this context? Seasoned players would probably not hesitate to call the cover system a "mechanic", something that connects players' actions with the purpose of the game and its main challenges. But the meaning of the term is not always clear.

During the summer of 2006, Nintendo released *Bit Generations*, a collection of seven games focused on minimalist game design. In *Orbital* (Nintendo, 2006), the player controls a small unit, flying between planets and meteorites. The goal is to collect items so that the initial particle grows until it has its own gravitational field, which can be used to attract a star and thus finish a level. The challenge is provided by the different gravitational fields of the other space bodies, and the fact that a crash with any stellar element will lead to the destruction of the player's unit. The player can only attract or repel her unit from these gravitational fields, and so use them as slingshots, safe havens, or u-turns.

Given this description, what are the mechanics of *Orbital*? A common answer could be the attraction/repulsion actions that the player can use, but also the gravitational fields of the planets or even the use of gravity for sling-shot flying. In this sense, then, game mechanics also describes the mechanisms of the game simulation. This lack of conceptual precision points to a definitional problem: it is unclear what game mechanics are, and how the term can be used in game analysis.

Game researchers and designers have provided a number of definitions of game mechanics that have been used in different contexts, from analysis (Järvinen, 2008) to game design (Hunicke, LeBlanc, Zubek, 2004). In this article, I propose a definition of game mechanics useful for the analysis of games and their formal constituents. This definition will allow for formalized analysis of game structures, and it will also open up for the possibility of connecting formal game analysis with research on controller designs and user experience.

I define game mechanics, using concepts from object-oriented programming, as methods invoked by agents, designed for interaction with the game state. With this formalized definition, I intend to:

- Provide a tool to discover, describe, and interrelate game mechanics in any given game.
- Define mechanics also in relation to elements of the game system, game hardware and player experience, mapping mechanics to input procedures and player emotions.

Even though I will be mentioning concepts like game rules, challenges, emotions and user experience, it is not my intention to enter the debate on those topics. Here, I use those concepts in a relational way: defining game mechanics requires mentioning and acknowledging rules, challenges and emotions. I do so in an instrumental way and leave for further research the implications of this definition for understanding other systemic components of games.

Since both game researchers and game designers have covered the topic of game mechanics, I begin this article with an analytical summary of the major works on this topic, providing a general overview of these previous definitions of game mechanics and place my work within this tradition.

The second part of this article presents the definition of game mechanics, detailing the elements that compose it. I then present a brief reflection on primary and secondary mechanics and how they can be derived from this definition.

These concepts are put into practice in the third part, where I perform a comparative analysis of *Shadow of the Colossus* (Team Ico, 2005), *Rez* (United Game Artists, 2002), and *Every Extend Extra* (Q Entertainment, 2006), highlighting the use of this concept of mechanics in the research on game structure and user experience. The article concludes with a summary of the results, and a reflection on the shortcomings of this definition.

With this article I intend to provide a practical analytical tool for describing game systems as formal structures that create gameplay. I also intend to focus on how variations in game design can innovate and deeply engage players in aesthetic experiences created by means of gameplay design.

Previous Definitions of Game Mechanics

There is a relatively long and multidisciplinary tradition of studying the ontology of games (Juul, 2005). The ontological question has often implied describing the elements of games, how players relate to these elements, and the contextualized act of play (*ibid*, p. 28). This study of games lead to analysis disregarding the overarching definitions of what games are and focused on each of the elements that constitute a game: the system, the player or the player-and-system in context. Eventually, this area of research was defined as game studies (Aarseth, 2001).

The research on games as systems lead to formal analysis of the game components and how they interrelate. Formal analysis is understood as descriptions of game components that can be discerned from others by means of their unique characteristics and properties. "Formal" should be understood in relation to aesthetic formalism, which contrasts "the artifact itself with its relations to entities outside itself" (Audi, 1999, p. 11).

Some formalist approaches makes a difference between the rules of the game and the actions afforded to players by those rules. This conceptual perspective can be tracked back to Avedon (1971) who suggests a formal structure of games in which there are "specific operations, required courses of action, method of play," which he defines as the "procedure for action", as opposed to the "rules governing action", which are "fixed principles that determine conduct and standards for behavior" (p. 422).

However, this formal distinction between rules and mechanics is not always applied in game mechanics research. Lundgren and Björk (2003) define game mechanics as "any part of the rule system of a game that covers one, and only one, possible kind of interaction that takes place during the game, be it general or specific (…) mechanics are regarded as a way to summarize game rules". In this view, mechanics is a term that encompasses those rules that are applied when the player interacts with the game, and there is no need for a definitional distinction between rules and mechanics. Game mechanics would be low-level descriptions of game rules or clusters of game rules.

Game designer Richard Rouse (2005) offers a more pragmatic approach to defining game mechanics, with the goal of teaching the basics of game documentation of game design. For Rouse, game mechanics are "the guts of a design document", since they describe "what the players are able to do in the game-world, how they do it, and how that leads to a compelling game experience" (p. 310). A similar pedagogical approach is taken by Fullerton, Hoffman and Swain (2004), who define "game procedures" (a similar concept to mechanics), as "the actions or methods of play allowed by the rules (…) they guide player behaviour, creating interactions" (p. 25). In teaching game design, then, there is a need to apply Avedon's conceptual distinction between rules and mechanics. The design process is understood as the creation of a system, and the interaction possibilities that a player has with that system. However, these approaches lack a deep explanation of the connections between rules and mechanics. These connections are fundamental for the formal analysis of games, as Björk and Holopainen (2005) stated in their argumentation for the development of Game Design Patterns.

Other definitions, like Cook's (2005): "game mechanics are rule based system/simulations that facilitate and encourage a user to explore and learn the properties of their possibility space through the use of feedback mechanisms", while acknowledging the relations between players, rules and mechanics, fail to provide a sufficiently clear set of properties that allows the concept to be applied in a formal analysis of games. This definition is valuable since it incorporates the notion of feedback to the understanding of mechanics, but it falls short in explaining how we can identify a mechanic, or a set of mechanics, and how it is based in the rule system.

The MDA Framework (Hunicke, Zubek, LeBlanc, 2004) provides some more detail on the formal nature of game mechanics: "mechanics describes the particular components of the game, at the level of data representations and algorithms (…) mechanics are the various actions, behaviours, and control mechanisms afforded to the player within a game context". The latter part of the definition provides a set of elements that will allow us to identify a mechanic. However, this definition would require more precision in its formulation: for instance, behaviours afforded to the player can be both strategies suggested by the game design (the level layout in *Gears of War* suggests the behaviour or covering, yet it does not directly afford that action); and the operations that the game system does in the background to calculate the success of player actions (as the effect of gravitational fields in *Orbital* - external to player agency, yet related with the player's actions).

The MDA framework provides insights into the relations between the formal, algorithmic elements of games and how they are presented to and manipulated by players. Nevertheless, it is a model that does not allow for the description and analysis of a mechanic due to a relative inconsistency in the formulation of the definition.

A much more precise approach is taken by Järvinen (2008), who not only distinguishes rules from mechanics but also relates the latter with player agency, both in terms of psychological and gameplay experiences. Järvinen defines mechanics as "means to guide the player into particular behaviour by constraining the space of possible plans to attain goals" (p. 254). In this sense, "game mechanics are best described with verbs" (p. 263), and so "take cover" would be a key mechanic in *Gears of War*, while the two dominant mechanics in *Orbital* would be "attract" and "repel".

In relation to rules, Järvinen perceives mechanics as making "a particular set of rules available to the player in the form of prescribed causal relations between game elements and their consequence to particular game states" (p. 254), which leads to the creation of player strategies derived from the intersection of rules and mechanics (p. 258).

Järvinen's approach is thorough, describing how players appropriate mechanics and how systems should be designed to afford strategy-generating mechanics. However, Järvinen's approach is rather deterministic: mechanics seem to exist so that goals can be achieved, and thus there would be no mechanics if the game, or a specific set of actions, has no goals. Cases like *Sim City* (Maxis, 1989) or some of the sandbox play in *Crackdown* (RealTime Worlds, 2007) encourage player action without the requirement of goals. Destroying a city by invoking Godzilla or exploring a sprawling postmodern metropolis using superhuman abilities are pleasurable interactions with(in) a game that are not determined by any in-game goal.

Within the general research tradition on game mechanics, the concept is used to analyze elements of the game system. Game mechanics are used to describe how players interact with rules, and as more formal properties of a game such as game goals, player actions and strategies, and game states. However, these definitions do not provide a single, dominant approach that encompasses all these aspects. All the previous definitions have attempted to provide pragmatic approaches to allow for a flexible understanding of game mechanics in games and how they relate to player agency and game rules. In the following section I present a formal definition of game mechanics, together with the arguments that make it a more precise and inclusive approach than those reviewed in this section.

Defining Game Mechanics

Let's start with a definition: game mechanics are methods invoked by agents, designed for interaction with the game state.

The different components of this definition require further explanation:

"Methods invoked by agents" defines this approach to game

mechanics, as it formalizes the use of terminology taken from the object oriented programming paradigm (Weisfeld, 2000). In this appropriation of the terminology, object orientation provides a set of metaphors that describe the elements of systems and their interrelations. I do not want to imply that the analysis of the source code of a game will reveal that all game mechanics have been implemented as methods of classes or that object-oriented programming should be considered a default methodology for the actual production of computer games. Nor am I implying that the Object Oriented Framework should be extended to a formal analysis of all elements of the game. Object Orientation provides a clear, formal framework for the description of games and as such is a useful analytical tool. It is useful because it provides a formalistic approach to actions taken within information systems like games, which may lead to the application of modeling languages like UML to the description of game systems. The Object Oriented framework is also appropriate because it facilitates an analysis that does not require human players to understand in-game agency. In other words, by using an Object-Oriented approach, we can analyze game mechanics as available both to human and artificial agents[1].

Following object oriented programming terminology, a method is understood as the actions or behaviors available to a class (Weisfeld, 2000, p. 13). Methods are the mechanisms an object has for accessing data within another object. A game mechanic, then, is the action invoked by an agent to interact with the game world, as constrained by the game rules. In *Gears of War*, if the player wants to take cover, she has to press the A button in the controller. This will make the avatar seek cover in the closest environment object that can provide that cover. In that sense, a mechanic is limited by the rules that apply to the gameworld (the general physics simulations, for instance, whose objects are suitable for providing some kind of cover), and, on occasion, to rules that apply exclusively to that particular mechanic - for example, some mechanics can only be invoked in certain environments or gameplay contexts.

Following Järvinen (2008), the best way of understanding mechanics as methods is to formalize them as verbs, with other syntactical/structural elements, such as rules, having influence on how those verbs act in the game. For example, in *Shadow of the Colossus* we find the following mechanics: to climb, ride (the horse), stab, jump, shoot (arrows), whistle, grab, run (and variations like swim or dive). In *Gears of War*, a non-comprehensive list would be: cover, shoot, reload, throw (grenade), look (at a point of interest), use, give orders, switch weapons[2]. All of these are methods for agency within the game world, actions the player can take within the space of possibility created by the rules.

This definition departs from the implicit anthropocentrism of previous approaches. Game mechanics can be invoked by any agent, be that human or part of the computer system. For instance, AI agents also have a number of methods available to interact with the gameworld. On occasion, those methods will be other than the ones made available to the human player, which can have consequences worth of analysis. This approach can be particularly interesting when trying to understand the effect of bots in MMORPGs, since bots are agents that optimize their interaction by focusing on a core set of mechanics. This design choice may lead to an imbalance in the game system, in terms of its dynamics or its economy. Another extension of this approach would draw a distinction between agents in a game with mechanics and agents without access to mechanics. For example, some bots do have access to mechanics while other game agents do not have access to mechanics and hence cannot interact with the game state. This line of research, however, is outside the scope of this article.

The second advantage is that it eases the mapping of mechanics to input devices, allowing for a great degree of granularity in the analysis of games. Applying the conceptual framework of Object Oriented programming determines that an agent invokes a mechanic in order to interact with the game[3]. When it comes to players, input devices - from mouse and keyboard to the Wii Fit Board - mediate this process. In *Gears of War*, the cover mechanic is invoked by pressing the A button in the controller. In *Orbital*, the two mechanics are mapped to the two buttons of the Game Boy Advance device. Thanks to the formal precision of Object oriented terminology, it would be possible to use an abstract modeling language, like UML, to describe the interaction possibilities afforded to players, and how those are mapped to specific input device triggers.

For game analysis, this suggests the possibility of closely studying the relations between input device design, and player actions. It would allow, for instance, the study of how in some fighting games, one mechanic is not triggered by one button, but by a combination of input processes. Thus, it could be argued from a formal perspective that mastery in fighting games comes from the mapping (Norman, 2002, pp. 17, 75-77), of one mechanic with a set of input procedures, which leads to both psychological and physiological mappings - how the "body" of a player learns to forget about the remembering the illogical sequence of inputs, and maps one mechanic to one set of coordinated, not necessarily conscious moves.

Another interesting approach from this formal perspective is the possibility of describing mechanics that are triggered depending on the context of the player presence in the game world, what I define as "context mechanics". In *Gears of War*, the cover mechanic depends not only on the specific input from the player, but also on the proximity of suitable objects to the player avatar. Contextual mechanics have also been used in *Assassins' Creed* (Ubisoft Montreal, 2007) to expand the possible interactions of the player with the gameworld, without overtly complicating the layout of the controller device.

Contextual mechanics are analytical concepts that can be used to understand how players decode the information in a level - how a player perceives certain structures and how those structures are used to communicate intended uses or behaviors. Furthermore, contextual mechanics can also be used to analyze a game like *Wario Ware, Inc., Mega Microgames!* (Nintendo R&D1, 2003) that builds its design by mapping multiple mechanics (Järvinen, 2008, pp. 266-269) to one button, easing the players' learning process and focusing on stress coping challenges (Rollings and Adams, 2007, pp. 287-288).

Implicit in this definition is an ontological difference between rules and mechanics. Game mechanics are concerned with the actual interaction with the game state, while rules provide the possibility space where that interaction is possible, regulating as well the transition between states. In this sense, rules are modeled after agency, while mechanics are modeled for agency.

In this object oriented framework, rules could be considered general or particular properties of the game system and its agents. All objects in games have properties. These properties are often either rules or determined by rules. These rules are evaluated by a game loop, an algorithm that relates the current state of the game and the properties of the objects with a number of conditions that consequently can modify the game state. For example, the winning condition, the losing condition and the effects of action in the player's avatar health are calculated when running the game loop. This algorithm relates rules with mechanics, exemplifying the applicability of an ontological distinction between rules and mechanics.

For example, in *Shadow of the Colossus* players have a game mechanic called "climb", but they are also determined by a property called "stamina", which is the algorithmic translation of a rule: "players have x stamina units". The climbing mechanic states that when invoked, stamina is lost at a certain ratio. A property/rule states that if stamina is below a certain threshold, climbing is not possible anymore. The game loop checks the game state; if the player invokes the climb mechanic, those functions that determine the consequences and boundaries of the players' interaction are called, and the resulting changes in the game state are evaluated against the rules of the game. Then, the player will succeed or not in "climbing", depending on their "stamina".

The second part of the definition claims that game mechanics are methods "designed for interaction with the game state". This implies that the task of game designers is to create mechanics that agents can use to interact with the game. These interactions modify the game state (Juul, 2005, 59-64). Game mechanics are often, but not necessarily, designed to overcome challenges, looking for specific transitions of the game state. Designers create the basic mechanics for the player correlating the central challenges of the game with the set of mechanics useful for overcoming them.

Challenges, like rules, are one of the contested areas in game research. Much has been written about what challenges are and how can they be analyzed, and it is not my intention to suggest a new interpretation of the term. In this article, I use challenge to refer to a situation in which the outcome desired by the player requires an effort to accomplish. For instance, every colossus in *Shadow of the Colossus* is a challenge, each of which is composed of a subset of challenges: the fifth colossus is a flying creature with weak spots in each wing and the tail. The challenge is to run from one weak spot to another without falling, since player movement is affected by the wind and the speed of the moving colossus. All these challenges are matched with a mechanic: by shooting arrows, the player calls the attention of the creature; by jumping and then grabbing to the hair of the creature, the player accesses a more or less stable surface where she can then run to the weak spots and stab them. All challenges in this example are mapped to particular game mechanics.

Even though this formal definition determines that games are structured as systems with mechanics, rules and challenges, understood as the essential grammar of computer games (and probably of all games), there is more to the act of playing a game than just interacting with mechanics constrained by rules. In the act of playing, players will appropriate agency within the game world and behave in unpredicted ways. One thing is what a designer previews, and another, very different one, is how players actually interact with the game world. The formal, analytical understanding of mechanics only allows us to design and predict courses of interaction, but not to determine how the game will always be played, or what the outcome of that experience will be.

Furthermore, it can happen that what was designed as a game mechanic is used in a non-gameplay related behavior: players of *Shadow of the Colossus* used the climbing mechanic to reach some of the farthest areas of the game world, which had no influence, or interest, for the central gameplay sequence and narrative of the game. Game mechanics are designed for gameplay, but they can be used for toyplay (Bateman and Boon, 2006). The only variation would be the level of abstraction: for a player who is playing the game, a mechanic serves a specific set of purposes, while a player that is playing with or within the game, a game mechanic loses its formal game design origin and becomes an instrument for agency.

For designers and theorists, game mechanics are discrete units that can be created, analyzed and put in relation to others. But for any agent in a game, the mechanics is everything that affords agency in the game world. Mechanics is thus tied to agency in the game system.

With this definition of game mechanics, I have intended to contribute to game studies by:

- Formalizing an ontological difference between rules and mechanics that can potentially lead to detailed game analysis, and
- Suggesting a mapping between game mechanics, input procedures, and player experience.

This very formal definition still leaves some questions unanswered, especially with regards to well-known terminology such as core mechanics. In the next section, I present some further implications of this definition for the analysis of games.

Interlude: Core, Primary and Secondary Mechanics

Game design literature uses the "game mechanics" concept extensively, incorporating certain qualifiers to it. It is not rare to find in the literature notions like "core mechanics" (Järvinen, 2998, p. 255; Rollings and Adams, 2007, pp. 316-357), and in more casual settings, an implicit categorization like primary mechanics and secondary mechanics (Järvinen, 2008, p. 268). These qualifiers do not describe what concept of game mechanics the authors are adopting - if a rule based one, in which mechanics is a subset of rules, or one that advocates for an ontological differentiation of both. In this section, I briefly discuss how core mechanics, primary mechanics and secondary mechanics can be used as functional terms within the context of the definition I have introduced. These concepts are, as said, widely used in game design literature, thus it is important to define them according to this article's definition of game mechanics.

Core mechanics, in the traditional sense, have been defined as "the essential play activity players perform again and again in a game (...) however, in many games, the core mechanic is a compound activity composed of a suite of actions" (Salen and Zimmerman, 2004, p. 316). Järvinen defined core mechanics as "the possible or preferred or encouraged means with which the player can interact with game elements as she is trying to influence the game state at hand towards the attainment of a goal" (255). Understanding core mechanics as those that describe the actions a player repeatedly performs is a useful formalism, but it falls short in precision. Players often perform play activities again and again in a game without using so called core mechanics. Jumping, for instance, is extensively used in multiplayer First Person Shooters, where almost all players spend some time "hopping" around - as a humorous display or for entertainment. Salen and Zimmerman and Järvinen are right in pointing out that the core mechanics have to do with repeated performance in the play context, but the actions performed ought to be defined from a systemic perspective, if the formal framework should be upheld.

From that systemic perspective, I define core mechanics as the game mechanics (repeatedly) used by agents to achieve a systemically rewarded end-game state. For instance, stabbing is a core mechanic of *Shadow of the Colossus*, since the player will perform it repeatedly to achieve the end state of the game, rewarded with the completion of the fictional framework of the game. In *Orbital*, the core mechanics are the only mechanics. Both games are examples of focused game design, in which player actions are limited, yet tuned to create emergent gameplay (Juul, 2005, pp. 67-83, Sweetster, 2008).

Games like *Sim City* or *EverQuest* (Sony Online Entertainment, 1999) do not have an end state as such. However, there are desired states towards which players focus their efforts, be those reaching the cap character level or keeping the city budget in the black. These games have a specific set of game mechanics oriented to reaching those states, and as such it is possible to speak of core mechanics even in the case of games with no systemically determined end state. In the case of simulations like *Sim City*, core mechanics are those that focus on reaching an equilibrium state; in games like *EverQuest*, core mechanics are those that allow players to reach a level cap, and further expand their agency by fine tuning their characters' abilities.

At this stage, readers will most likely object that complex games like *Grand Theft Auto IV* have such a vast number of mechanics, and so many are used to make the game progress, that the very use of the core mechanics concept may be useless. It is a valid point - complexity requires a precise terminology. Thus, I will use the concepts of primary (core) mechanics and secondary (core) mechanics to solve some of these issues.

The concept of primary mechanics has been defined by Järvinen (2008, p. 268) as "what the player does in relation to a game state during a standard turn or sequence", differentiating then between submechanics, or actions available to the player "as a consequence of the primary mechanic" (*ibid*), and modifier mechanics, or actions the player does "in a specific game state which occurs on some condition (…) specified in the rules" (*ibid*). Again, Järvinen's comprehensive approach is highly relevant, but its formal ties to games understood as goal-oriented systems with which (human) agents interact determine this classification of mechanics. In the following I will suggest an approach to the concepts closer to the approach taken in this article.

Primary mechanics can be understood as core mechanics that can be directly applied to solving challenges that lead to the desired end state. Primary mechanics are readily available, explained in the early stages of the game, and consistent throughout the game experience. In *Grand Theft Auto IV*, primary mechanics are shooting, melee fighting, and driving: they are readily available to the player, mapped to the most obvious and tradition-conforming controller inputs and remain consistent throughout the game experience: shooting is always performed using the same button combination, and when players have control, they always have access to that mechanic, provided they have a firearm. Interestingly, this use of the primary mechanics concept explains the design experiment of *Orbital:* players only have primary mechanics available to interact with the gameworld.

Secondary mechanics, on the other hand, are core mechanics that ease the player's interaction with the game towards reaching the end state. Secondary mechanics are either available occasionally or require their combination with a primary mechanic in order to be functional. The cover mechanic in *Grand Theft Auto IV* is an example: it cannot be used exclusively to solve the main challenges of the game, but once mastered, it can prove of help to achieve the end state of the game. In comparison, the cover mechanic of *Gears of War* is primary, since not using it implies the almost immediate death of any game agent.

Again, readers may claim that there are mechanics in a game beyond those tied to the goal/reward structure. And they are right - in many modern, complex computer games there are many mechanics available for player agency, and several of them play a role in achieving the goals. I would prefer not to categorize those, though: the importance of the terms of primary and secondary is their explanation of the game system as it was intended to be played by an ideal player[4]. Any formalist approach, such as the one proposed in this article, falls short of trying to explain all possible player interactions. As such, I would like to leave all mechanics that cannot be consistently defined as primary or secondary without any type of classification. It is still relevant to understand them and to describe how their importance is perceived in actual gameplay. However, those goals are beyond the scope of this article.

The distinction between primary and secondary, then, allows for a granular understanding of the agency methods available for players in the game experience, and their importance in terms of design and analysis. However, these terms should not be used as rigid categories: on occasions, secondary mechanics can turn into primary mechanics during the designed gameplay progression, and some primary mechanics may even disappear in the length of a game. These concepts should be used as analytical aids, as a first step into a formal categorization of mechanics depending on their impact on gameplay.

One last question remains: within this formal, object oriented framework, it is not possible to describe systems like the driving mechanic in *Grand Theft Auto IV*: more precisely, driving would consist of braking, accelerating, steering and hand-breaking. All of these are, effectively, the methods invoked by agents in order to interact with the game. However, using this very detailed description is not always a useful approach. Thus, the concept of compound game mechanics can be of use: a compound game mechanic is a set of related game mechanics that function together within one delimited agent interaction mode. These modes are defined by the interaction of these different modalities: as such, the driving compound mechanic is composed by a set of mechanics interrelated to provide a relatively accurate model of driving. When playing, and, on occasion, when analyzing, it is useful to think about these compound mechanics as a whole and not as a collection of formally differentiated mechanics. This concept provides an appropriate shelter for those complex interaction processes that, while composed by a number of smaller formally determined mechanics, we as players, analysts and designers, think of as unified.

So far, this article has been a rather dry presentation and argumentation for a terminological, analytical position. In the next section I will apply this definition, with attention to input-interface configuration and plausible player experience, to the analysis of the common mechanics and effects of *Rez*, *Shadow of the Colossus* and *Every Extend Extra*.

Applying the Definition: Theory and Design

To prove the analytical use of my definition of game mechanics, I apply it to three different games. This application will show that game mechanics can be used not only to formally describe a game but also to thread connections between different games and intended player experiences. In the following examples, I trace such a connection between *Shadow of the Colossus*, *Rez* and *Every Extend Extra* by analyzing dominant game mechanics and their implementation, and interpreting how the design choices could be meant to affect the player experience.

The basic mechanic in *Shadow of the Colossus* can be called "stabbing", which requires players to select a specific weapon when placed in a specific spot of a colossus, then press once the x button to "charge" her attack, then press once again to release and effectively stab the colossus. The intensity of the attack depends on the time lapse between the two inputs: the longer the player waits to unleash the attack, the more damaging it will be.

From a purely analytical perspective, this mechanic introduces an interesting observation: as opposed to the more classical "aggression" mechanics, in *SoTC* players do not obtain direct output from their initial input, nor do they have to push down the button for "charging" the attack. This is arguably a design choice, and it could be tied to the aesthetic goals of the game: the player is in a weak position between inputs, which reinforces the sense of awe these colossi suggest. In many computer games, players are supposed to feel empowered, yet challenged by their enemies. *SoTC* is designed to present players with what appears like an insurmountable enemy and equips them with just the bare abilities to epically undergo the slaying of these creatures.

By slightly modifying a well-known game mechanic, it could be argued that the design of *Shadow of the Colossus* is intended to create an experience of powerlessness and epic achievement. The player is not only faced with the colossi as challenges, but also their repertoire (Juul: 2005) is challenged by the control configuration of the attack mechanic. This challenge has likely been designed to have a significant emotional impact on the player, which I will analyze at a later stage in this section.

Even though this analysis could itself justify the use of this formal definition of game mechanic, it also allows for extending the study of mechanics to comparative approaches. In the rhythm shooter *Rez*, players have a general mechanic "shoot" that is invoked as follows: while holding the x button, players can select enemies with their crosshair, up to a limit of 8. When releasing the x button, players destroy the enemies. For each enemy destroyed, a rule states that a beat is played, hence the rhythm-based gameplay of the game.

From this brief description, we can argue that there are similarities between the two mechanics, as they both modify the conventional input/output mechanic: instead of pushing a button to produce an output, players have to release it to produce the output. The analysis can be extended: there is a principle of tension and release at work both in the stab mechanic of *Shadow of the Colossus* and in the shoot mechanic of *Rez*, and both can be interpreted as design choices that create a specific player experience.

Music can sometimes be structured as harmonic periods of tension and release: a composition builds up to a moment where the chord progression, for example, is culminated in a tonal change or a different tempo (A more detailed explanation of the structure of music and how it can be interpreted in the context of technological experience can be found in McCarthy and Wright, 2006). The same principle dominates *Rez*: players build up tension by targeting multiple enemies, then releasing and creating music beats. And in *Shadow of the Colossus,* players experience tension while their stabbing "strength" is being loaded and release when the player hits the button to stab the colossus. By examining the formal properties of these two mechanics, we can argue for a connection to an intended player experience, which means that it is possible to recognize patterns or typologies in the design of mechanics.

This tension and release effect through mechanics can also be found in the game *Every Extend Extra*[5], where the main mechanic "to explode" is executed by pressing the x button. This input makes the avatar explode and start a chain reaction rewarded with points. Tension is created by avoiding collision with the incoming enemies, which would destroy the player avatar without causing a chain reaction, while waiting for the perfect combination of enemies onscreen that would allow for a large chain-reaction effect. Gameplay is built around the exploding mechanic, another tension-release mechanic type: tension is built while avoiding enemies without providing any input, and release comes when the player finds the right timing and place to trigger the explosion.

These three reasonably different games are connected by a similar interpretation of a game mechanic. All these games play with player expectations (action on release) with the intention of creating a specific emotional experience in players. In the case of *Shadow of the Colossus* the experience is associated with the excitement of attacking the colossi with maximum power without falling. In *Rez* and *Every Extend Extra*, it could be argued that the synaesthetic goal of these games is communicated also by means of the mechanic: players experience the musical tension and release structure while actually playing the game.

From a formal analytical perspective, there is a connection between *Shadow of the Colossus*, *Rez* and *Every Extend Extra*, since all this games have manipulated a well-known core mechanic into a process based one of tension and release. This connection also leads to a plausible relation between the design of these mechanics and its possible impact on the player experience. By modifying the player expectations, and meaningfully changing the input procedures, these games are intended to create emotional experiences based on the agency of players with the game state and how it reacts to their input.

By tracing relationships between game mechanics, and arguing for their intended effects on players, game designers may innovate their approach to agency through the design of the game system. It could be argued that the developers of the three aforementioned games did so by formally isolating the basic processes of those mechanics, partially altering them, consequently modifying player expectations and experience.

As I have already hinted at, game mechanics are not only formally recognizable by designers; they are also a big part of the players' repertoire (Juul, 2005, p. 97-102). By modifying the basic interaction patterns of a mechanic, designers can arguably expect to break player expectations. A possible use of this definition, then, is as a formal tool for describing and modifying mechanics in a coherent and comprehensive way, by understanding the relations between the different methods, its properties, and how those are mapped onto the control interface.

Another potential contribution to game design is related to its documentation and communication. When writing a design document, game designers often have to translate into words their ideas about player interaction with the game world how that interaction is constrained by rules and how those mechanics can help overcoming the challenges in interesting ways. The literature on game documentation is vast (Rollings and Adams, 2007, pp. 63-65; Rouse, 2005, pp. 355-381, Fullerton, 2008, pp. 394-412, Schuytema, 2007, pp. 83-116), and most of it is based on tradition or a set of common practices more than on a research-based approach to the formal elements of games. With this definition of game mechanics, designers could more easily translate their ideas into a formal set of methods (mechanics), properties (rules that define the scope of those mechanics) and challenges.

Finally, for design and development purposes, this definition's focus on an object-oriented approach can facilitate the communication between programmers and designers with limited technical background. By thinking about rules and mechanics as designed methods and properties, game designers could perhaps document and explain their concepts with more precision, enhancing productivity while creating more comprehensive documentation for game development.

Conclusion

This article was born out of necessity: having an analytical vocabulary for defining game structures and systems that allowed a formal, precise, and scalable description of games as systems and how they interrelate with player practices. The result of this necessity is a formal definition of game mechanics that owes to object-oriented programming its formal phrasing, while inheriting from game studies the figure of players, or agents, as fundamental to understand how games are designed and played.

This article has defined game mechanics as methods invoked by agents for interacting with the game world. This definition allows the study of the systemic structure of games in terms of actions afforded to agents to overcome challenges, but also the analysis of how actions are mapped onto input devices and how mechanics can be used to create specific emotional experiences in players.

There are, however, many grey areas I do not have the space to focus on here. Perhaps the most significant is the ontological distinction between rules and mechanics. Game researchers have argued convincingly that mechanics could be understood as subsets of rules. However, rules are normative, while mechanics are performative, and I have argued that this ontological distinction can be extremely beneficial for the analysis of computer games.

Game studies history shows that there is no dominant definition of key concepts like rules or mechanics, and that those that attempted have yet to succeed. This article should not be read as the ultimate definition of game mechanics. This definition is flawed, yet less so than some previous ones. My goal will be achieved if I have succeeded in communicating to the reader one simple notion: that it is possible and useful to understand game mechanics as different from game rules, and in that understanding, we can more clearly describe how games can be designed to affect players in unprecedented ways.

Acknowledgements

The author would like to express his gratitude to the anonymous Game Studies reviewers who offered constructive and illuminating feedback, and to Aki Järvinen, Jesper Juul and Olli Leino, who helped shape earlier versions of this article with their comments.

References

Aarseth, E. (2001). "Game Studies: Year One". *Game Studies*. Volume One, Issue One. Retrieved September 22nd, 2008.

Abelson, H. & Gerald Jay Sussman with Julie Susman (1985). *Structure and Interpretation of Computer Programs*. Cambridge, Massachusetts: The MIT Press.

Adams, E & Andrew Rollings (2007). *Fundamentals of Game Design*. New Jersey: Pearson-Prentice Hall.

Audi, Robert (ed.) (1999). *The Cambridge Dictionary of Philosophy*. Cambridge: Cambridge University Press.

Avedon, E. M. (1971). "The Structural Elements of Games". In Avedon, E. M. & Brian Sutton-Smith (eds.) (1971), *The Study of Games*. New York: John Wiley and Sons.

Bateman, C. & Richard Boon (2006). *XXI Century Game Design*. Hingham, Massachusetts: Charles River Media.

Björk, S. & Jussi Holopainen (2005). *Patterns in Game Design*, Hingham, Massachusetts: Charles River Media.

Cook, D. (2006). "What are game mechanics?". lostgarden.com, available online at http://lostgarden.com/2006/10/what-are-game-mechanics.html (accessed: 26/3/2008).

Dillon, A. (1995). "Artifacts as theories: Convergence through user-centered design". In Kinney, T. (ed.), *Proceedings American Society for Information Science*, 32, pages pp. 208-210.

Floridi, L. & Jeff Sanders (2004), "Levellism and the Method of Abstraction". Information Ethics Group Research Report. Available at http://web2.comlab.ox.ac.uk/oucl/research/areas/ieg/research_reports/ieg_rr221104.pdf (accessed: 26/3/2008).

Fullerton, T. & Christopher Swain & Steven Hoffman (2004). *Game Design Workshop. Designing, Prototyping and Playtesting Games*. San Francisco: CMP Books.

Fullerton, Tracy (2008), *Game Design Workshop. Second Edition. A Playcentric Approach to Creating Innovative Games*. New York: Morgan Kaufmann.

Hunicke, R. & Marc LeBlanc & Robert Zubek (2004), "MDA: A Formal Approach to Game Design and Game Research". Available online at http://www.cs.northwestern.edu/~hunicke/MDA.pdf (accessed: 26/3/2008).

Iser, W. (1980). *The Act of Reading. A Theory of Aesthetic Response*. Baltimore: The Johns Hopkins University Press.

Järvinen, A. (2008). *Games without Frontiers: Theories and Methods for Game Studies and Design*. Tampere: Tampere University Press. Available at http://acta.uta.fi/english/teos.phtml?11046 (accessed: 26/3/2008)

Juul, J. (2005). *Half Real. Videogames between Real Rules and Fictional Worlds*. Cambridge, Massachusetts: The MIT Press.

Lundgren, S. & Staffan Björk (2003). "Game Mechanics: Describing Computer-Augmented Games in Terms of Interaction". In *Terms of Interaction. Proceedings of TIDSE 2003*, pp. 45-56. Available online at http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.5147 (accessed 1/9/2008).

McCarthy, J. & Peter Wright (2006). *Technology as Experience*. Cambridge, Massachusetts: The MIT Press.

Norman, D. (2002). *The Design of Everyday Things*. New York: Basic Books.

Rollings, A. & Dave Morris (2004). *Game Architecture and Design: A New Edition*. Indianapolis, Indiana: New Riders Press.

Rollings, A. & Ernst Adams (2007). *Fundamentals of Game Design*. New Jersey: Pearson Prentice Hall.

Rouse III, R (2005). *Game Design Theory and Practice*. Plano, Texas: Wordware Publishing Inc.

Salen, K. & Eric Zimmerman (2004). *Rules of Play. Game Design Fundamentals*. Cambridge, Massachusetts: The MIT Press.

Schuytema, P. (2008). *Game Design. A Practical Approach*. Boston: Charles River Media.

Sweetster, P. (2008). *Emergence in Games*. Boston: Charles River Media.

Weisfeld, Matt (2000). *The Object Oriented Thought Process*. Indianapolis, Indiana: Sams Publishing.

Ludography

Epic Games (2006), *Gears of War* (Xbox 360)

Matsuhisa, Kanta (2004), *Every Extend* (Windows)

Maxis (1989), *Sim City* (Windows)

Namco (2003), *kill.switch* (PlayStation 2)

Nintendo (2006), *New Super Mario Bros*. (Nintendo DS)

Nintendo (2006), *Orbital* (GameBoy Advanced)

Nintendo R&D1 (2003), *Wario Ware, Inc.: Mega Microgames!* (GameBoy Advanced)

Q Entertainment (2006), *Every Extend Extra* (PSP)

RealTime Worlds (2007), *Crackdown* (Xbox 360)

RockStar North (2008), *Grand Theft Auto IV* (Xbox 360)

Sony Online Entertainment (1999), *EverQuest* (Windows)

Team Ico (2005), *Shadow of the Colossus* (PlayStation 2)

Ubisoft Montreal (2008), *Assassins' Creed* (Xbox 360)

United Game Artists (2002), *Rez* (PlayStation 2)

Endnotes

[1]It is possible to find other applications of Object Oriented modeling to the study of computer games. For instance, the concept of Inheritance, or how some classes are derived from preexisting classes, can be used to explain different mechanics available to different agents in a gameworld. Other uses of the Object Oriented framework in the analysis of information systems can be found in Floridi and Sanders (2004).

[2]Järvinen (2008) has a detailed list of all the mechanics, understood as verbs, present in the micro-game collection *Wario Ware*. My approach is deeply inspired by that listing.

[3]In the case of analyzing mechanics as available to artificial agents (i.e. A.I. controlled bots), it is possible to disregard the mapping between mechanics and input controllers.

[4]Even though the use of the "ideal player" here can invoke literary theory approaches to the ideal reader (Iser: 1980, pp. 27-30), I will be using "ideal player" in a more design-oriented perspective, as the abstraction of a user that will use the object designed as predicted by the design team (see Dillon: 1995).

[5]*Every Extend Extra* is the PSP version of an earlier game built with the same mechanics, *Every Extend* (2004)