PLAYABILITY IN ACTION VIDEOGAMES: A QUALITATIVE DESIGN MODEL

CARLO FABRICATORE, MIGUEL NUSSBAUM AND RICARDO ROSAS

ABSTRACT

In the 1990s the videogame industry has managed to become the fastest growing segment of the entertainment industry in America. However, only a very low number of videogame products manage to cover the costs of production and generate earnings. According to traditional marketing wisdom, players' preferences are a core issue in creating successful products, and the game design process is crucial for guaranteeing players' satisfaction.

Then, an important question arises: what do players want in videogames?

The purpose of this work is to propose a game design reference that directly mirrors players' preference, shaped as a qualitative model based on empirical data gathered during playing sessions. The model describes the main elements that, according to players' opinions, determine the playability of action videogames, and proposes design guidelines that are the conceptualization of players' preferences. Therefore, the model helps game designers to understand the elements that must be dealt with in order to make better games.

Besides the operational relevance of the model, the research methodology described in this work is an example of how a qualitative approach such as the Grounded Theory paradigm can be applied to solve a software specification problem directly focusing on end-users.

CONTENTS

1 INTRODUCTION

1.1 What Do Players Want in Videogames?

In the 1990s the videogame industry has managed to become the fastest growing segment of the entertainment industry in America (Rinaldi, 1998). However, due to the high costs of production, only very few products sell enough to generate earnings. As a reference, suffice it to consider that an estimation of 1996 sales indicates that only 4% of the products commercialized that year covered the costs of production and generated earnings, and things haven't changed much since then (Business Week, October 1997).

According to traditional marketing wisdom, the success or failure of a product depends mainly on how well it satisfies customers' preferences, needs and expectations (Kotler, 1993). Consequently, in the case of videogames, knowing the player and his/her preferences is crucial to design products capable of satisfying their target market.

This context led us to formulate our research questions: what, according to player's preferences, determines the quality of a videogame? What do players want in videogames?

To find an answer to these questions, we focused on single-player games and individual playing experiences.

1.2 Videogame Design in Existing Bibliographic References

We started our research by reviewing bibliographic sources related to videogames, to see how they tackle the issues of quality and players' preferences. We found that there is no great abundance of sources regarding these issues, and that the main contributions come from the fields of educational videogames, and pure-entertainment videogame design.

As for the field of educational videogames, we found very interesting results in T. W. Malone and M. R. Lepper's works (Malone, 1981a, 1981b; Malone & Lepper, 1987), whose focus was mainly set on providing a taxonomy of elements that make computer games intrinsically motivating (i.e. without the intervention of exogenous factors). The authors identify two classes of motivating factors: individual and interpersonal.

The latter were not considered relevant to our research, due to our single-player focus. Apropos individual factors, in the alluded references the authors posit that intrinsic motivation in games is mainly determined by four elements: challenge, curiosity, control and fantasy.

A game should provide challenges of intermediate difficulty for the player. Important elements that determine the level of motivation of the challenge are: goals, uncertainty of the outcome, and performance feedback. In order for the challenge to be motivating, goals must be clearly defined, and hopefully organized in a hierarchy relating both short-term and long-term objectives. The outcome of the game should be uncertain, using variable-difficulty levels, multiple levels of goals, hidden information and randomness in order to challenge while at the same time avoiding both triviality and nearly impossible difficulties. Finally, in order to sustain motivation, performance feedback should be clear, constructive and encouraging, which will also contribute to player's self-esteem.

Curiosity should be stimulated by an optimal (moderate) level of information complexity, or discrepancy with the individual's present expectations and knowledge. During the playing experience, curiosity can be triggered by means of variability in audio and visual effects, and situations that intrigue and surprise players.

Games should also promote feelings of self-determination and control. These are determined by three elements: contingency, choice and power. As for contingency, the outcomes of the games should be contingent (i.e. directly dependent) upon players' responses. The game can also enhance the sense of control by providing a moderately high level of choices over various aspects of the playing activity. Finally, the possibility of producing powerful effects in the gaming environment is another control-enhancing feature.

As for fantasy, the authors define a fantasy environment as one that evokes mental images or physical or social situations not actually present, and speak about exogenous and endogenous fantasies. In the former, the fantasy serves as a mere frame for the real playing activity.

Endogenous fantasies are instead intrinsically related to the activity, being its goals and dynamics strictly determined by the characteristics of the gaming world. These fantasies are considered the best ones to promote intrinsic motivation in games.

The results of Malone and Lepper's research (1987) were drawn complementing empirical studies involving children playing educational videogames with the analysis of motivation theories. The generality of such theories makes the authors' conclusions regarding intrinsic motivation relevant even for mainstream pure-entertainment products. Additionally, the empirical component of their approach is a guarantee that their results mirror children's direct preferences. However, we couldn't take Malone and Lepper's results (1987) as a complete answer to our questions (and they were very probably not intended as such), at least because the group of players and the category of videogames analyzed by the authors are not representative of the mainstream pure-entertainment products and players.

The field of videogame design provides sources explicitly focused on supporting the creation of better pure-entertainment products (Costikyan, 1994; Crawford, 1982; Lewinski, 1999; Rouse, 2001; Saltzman, 2000).

We found these sources to be the most relevant to understanding what players like in videogames. More specifically, we considered R. Rouse's work (2001) representative of current game design wisdom, since it summarizes fairly what many other authors wrote about players' preferences. Rouse speaks about elements that motivate players to play a game, and expectations that players have when they are actually playing. Setting the focus on single-player games, the motivating factors are:

1) The challenge.
2) A dynamic solitaire experience, wherein they can be in control of an interactive environment.
3) Sense of achievement and self-satisfaction.
4) An emotional experience, whose complexity can vary from game to game, but which should always guarantee an emotional payoff as a reward for playing the game.
5) A good fantasy, providing an opportunity to see the world through someone else's eyes, or take part in otherwise inaccessible experiences.

As for expectations, Rouse posits that players expect:

1) To interact with a consistent world, being able to understand what actions they are allowed to perform, and what results those actions will produce.
2) To understand the bounds of the game, since even though players don't have to understand from the very beginning which actions are possible and which are not, they should immediately understand what is outside the scope of the game.
3) Reasonable solutions to work, which often implies offering the player the possibility of choosing between alternative solutions for a specific problem.
4) Direction, ensuring that players always have an idea of what their goal is.
5) To accomplish a task incrementally, knowing that they are on the right track by having clear checkpoints (or sub-goals) along the way, which is also a means of providing short-term rewards.
6) To be fully immersed in the playing experience, without disruptive elements that could shatter the suspension of disbelief.
7) To fail, since players tend not to enjoy trivial challenges.
8) A fair chance, since players don't like to feel that they are facing obstacles or opponents which require too much trial and error or are too difficult to overcome.
9) Not need to repeat themselves, and avoid having to accomplish the same goal over and over again. This, unless the whole game is built around this concept (as is the case of many sports games, in which the goal is always the same and the condition to win is achieving it as many times as possible); or if the reward changes each time that the goal is achieved.
10) Not to get hopelessly stuck, reaching dead-ends that force the gaming experience to be started again from the beginning.

11) To act, not to watch, stressing the importance of players' active role in the playing experience, which implies the need to minimize the number and length of instances in which they become mere spectators.

Besides dealing with motivation and players' expectations, the reviewed references propose operational design information to enhance the possibilities of satisfying players, tackling specific topics such as the gameplay (Howland, 1998; Rouse, 2001), interactivity (Crawford, 1988), storytelling (Hanscome, 1995; Lewinski, 1999; Rouse, 2001), usage of AI (Rouse, 2001; Saltzman, 2000) and the importance of character development (Spector, 1998). This information, just like the principles regarding motivation and expectations, is based on the authors' experience in the field of game design, and includes neither a systematic description of direct players' preferences, nor principles directly drawn from them, thus lacking the empirical component of Malone and Lepper's approach. This motivated us to try to complement the available sources focusing on the mainstream videogame audience and pure-entertainment products, and following an empirical approach. Hence, we decided to conduct a qualitative study based on empirical data mirroring direct players' preferences, with the goal of learning from players what makes good games, and finding systematic and rigorous results to support the game design process.

1.3 Scope of the Research

To understand the fundamental determinants of videogame quality as viewed by players, we selected a specific game genre to be studied, since design aspects' importance may vary from genre to genre. We reviewed 1997-1999 monthly issues of the Computer Gaming World (CGW) magazine's *top-100 monthly readers' poll*, and identified action, strategy and sport as the three most played genres, with no clear winner among them. Hence, we selected the action genre based on popularity criteria and on its historical relevance (the first commercial hits, such as Namco's `Pac-Man`, and Nintendo's `Donkey Kong`, were all action games).

After the selection of the genre, we further narrowed the scope of the research based on the importance of the two types of information managed by players during the game-playing: functional and ambience information.

Ambience information encompasses merely perceptual elements that contribute to creating a specific atmosphere capable of drawing and maintaining players' attention on an emotive basis, making them feel part of the gaming world. Consequently, ambience information is determined by the fantasy of the game, which makes it difficult to develop general ambience guidelines valid for all games.

Functional information allows the player to understand and control the gameplay, which is the set of all that can be done by the player in the gaming world (Howland, 1998; Lewinski, 1999). Thus, functional information is what allows players to actually interact with the gaming world, and without it there would be no game-playing at all.

Playability is the instantiation of the general concept of usability when applied to videogames, and it is determined by the possibility of understanding and/or controlling the gameplay. Poor playability cannot be balanced by any non-functional aspect of the design, since a very good gaming atmosphere by itself means nothing if the player can't understand and play the game.

All this led us to focus our research exclusively on the issue of playability in action videogames.

2 METHODOLOGY

2.1 Methodological Framework

Finding the determinants of the playability of an action videogame as viewed by players implies an empirical study of persons' experiences with a specific phenomenon. Therefore, to conduct our research we adopted the *Grounded Theory* method, which provides procedures designed to allow researchers to develop theoretical formulations to describe and explain specific phenomena, relying merely on qualitative empirical evidence (Strauss & Corbin, 1990).

According to this method, the researcher doesn't formulate hypotheses, trying later on to test them against empirical data. He/she rather works on an emerging theory constantly revised and eventually modified during the research as new important patterns, concepts or relationships emerge from the analysis of data (Glaser & Strauss, 1967).

The method provides a systematic procedure that integrates sampling, data gathering and analysis in iterative cycles. The procedure begins with the definition of the objective of the research (represented by research questions). Then, a sample of experimental subjects is chosen, and data is gathered and analyzed. The analysis can lead to conclusive or partial results. If the results are not conclusive, they could require gathering more or different data, or modifying the sample, consequently initiating a new cycle of data gathering and analysis. More precisely, the procedure relies on the criterion of *theoretical saturation*, which posits that it is necessary to sample until the examination of new data reveals no new information regarding the conceptual categories analyzed and their relationships. Once such criterion is satisfied, the results of the analysis can be considered conclusive and inductively generalized to the universe represented by the selected sample (Glaser & Strauss, 1967; Strauss & Corbin, 1990).

2.2 Data Collection

Data was collected during and after individual playing sessions by observing and logging users' activities, and gathering users' opinions by directly interviewing players.

During the playing sessions we used 39 action videogames, selected according to the 1997-1999 CGW's *top-100 monthly readers' poll*. To ensure their relevance only games ranked for more than three months in the *top-100 poll* were considered eligible.

The subjects involved in the research were 53 male Chilean players, aged between 20 and 30. The size of the sample was determined by the application of the principle of theoretical saturation (Strauss & Corbin, 1990). The range of ages of the participants was chosen based on the average age of CGW readers (according to the available data, 40% of them are aged between 25 and 34 (G. Jones, personal communication, June 1999). As for the gender, only male players were selected due to the low availability of female Chilean players, and to the impossibility of explicitly testing whether preferences regarding videogames are gender-independent or not.

In order to guarantee the expertise of the participating players, and taking as a reference *PC Data's* 1997 *Computer Gaming in America* survey (which defined core gamers as playing 26 games per year, and casual around 4 per year) (Wilson, 1998), the sample included only people playing at least 15 games per year.

Each playing session was preceded by an introductory explanation of the aim of the study. This ensured that players focused their comments and observations on playability issues regarding the games they were going to play, and that they provided opinions and recommendations about the quality of the design, and how it could be improved.

Players were free to play until they felt they knew the game enough to discuss it and express judgments. Playing sessions ranged from 1.5 to 2.0 hrs, with an average duration of 1.9 hrs.

During each session, spontaneous comments regarding the issue of playability were noted or tape-recorded. After the session, the topic of playability was further discussed to focus on specific issues that, according to the researcher's opinion, received only a superficial treatment in spite of their importance.

2.3 Data Analysis

For the analysis of the raw data (i.e. players' sentences), we followed the procedure summarized below.

1) Break down and conceptualize raw data
2) Categorize concepts
3) Describe categories
4) Analyze relationships between categories
5) For each category, analyze players' preferences regarding playability

First, raw data was analyzed to conceptualize, label and categorize design issues that players considered important determinants of the playability of a game. Then we described each category, relying on empirical data and the analysis of sampled games. After that, we studied relationships between categories, based on statements of relationship explicitly made by the players or implicit in their comments. Finally, for each category we analyzed players' preferences, deducing a set of playability design guidelines classified by players as prescriptions (to be followed in order to design playable games), and recommendations (considered playability enhancers, although not following them would not compromise the overall playability of a game).

Inconsistencies in the results (i.e., conclusions not every player agreed on) and incomplete information (i.e. concepts brought up by only a few players) were tackled by means of focus groups, with the researcher acting as moderator. The result was a general agreement on the functional relevance of all the guidelines, with sporadic disagreement on whether some of them should have been considered prescriptions or recommendations. In these cases we followed the opinion of the majority.

For each step of the procedure we carried out specific actions, usually driven by guiding questions. Below we describe further details of the procedure, exemplifying whenever possible its application by using the extracts of raw data in Table 1.

Table 1: Examples of raw data

Extract N. 1. Game: `Grand Theft Auto`
"…when I go under the bridge, I don't see my car anymore! Luckily, it's not for too long…"
Extract N. 2. Game: `Unreal`
"…I was scared by this guy who suddenly popped into the room. He was a *Nali*, and they are friendly, but I didn't recognize him from a distance. I thought he was an enemy because his appearance doesn't look too friendly, and I killed him! Argh! You should always be able to recognize your friends… Too bad…"

2.3.1 Breakdown and Conceptualization of Raw Data

Goal

Analyze players' statements searching for key gameplay concepts, i.e. those related to the ability to play the game.

Actions

1) Find explicit concepts, referred to in the statements by means of nouns or short definitions that identify them in a clear way.
2) Find implicit notions underlying players' statements, and label them as new concepts.

Guiding questions

- What is the player talking about?
- Does it regard his ability to play the game?

Example

In the first extract the player talked about his car disappearing under a bridge. In this case <u>car</u> and <u>bridge</u> are explicit gameplay concepts, since they are key to an event that affects the ability to play (the car disappearing under the bridge), as proved by the fact that, in latter statements, the player asserted that such event prevented him from seeing the car, and that luckily it didn't last too long. Additionally, the event is clearly related to the notion of <u>point of view</u>, therefore considered as an implicit gameplay concept.

In the second extract the player referred to a friendly entity that appeared unexpectedly in the scene, and was mistaken for an enemy and consequently killed. This event regards the concepts <u>guy</u>, <u>enemy</u>, and <u>friendly</u>. These notions can be considered gameplay concepts since they affected the player's ability to play, given that, in latter statements, he judged the whole event as something that should not happen. Additionally, in his second statement the player referred to the alluded entity as being a *<u>Nali</u>*, which is another gameplay concept, since the player complained explicitly about not being able to identify the creature immediately.

This analysis leads to the set of concepts reported in Table 2.

Table 2: Conceptualization of data

Source	Data	Explicit concept(s)	Implicit concept(s)
Extract N. 1	"When I go under the *bridge*, I don't see my *car* anymore!"	Car, bridge	Point of view
Extract N. 2	"…I was scared by this *guy*..."	Guy	
Extract N. 2	"He was a *Nali*, and they are friendly…"	Nali, friendly	
Extract N. 2	"I thought he was an *enemy*…"	Enemy	

2.3.2 Categorization of Concepts

Goal

Categorize concepts, based on properties explicitly referred to by players and/or deduced from the observation of the playing experience and the analysis of reviewed games.

Actions

1) Identify concepts' properties.
2) Compare concepts based on their properties.
3) Define categories according to similarities between concepts.

Guiding questions

- What are the properties of this concept?
- What are its similarities with other concepts?

Example

Let's start by considering the concepts car and guy. From the observation of the playing sessions, we saw that in `Grand Theft Auto` the car the player referred to was something that he could control. However, in the same game there were other, non-player-controlled instances of the same concept. Plus, we observed that in that game all the cars interacted with each other and with the scenario, and that certain interactions could damage cars, possibly destroying them. In `Unreal`, the guy mentioned by the player was a humanoid non-player-controlled creature. The creature could interact with the environment, had a limited amount of vital energy, and if wounded too many times he would die. We found that in the game there were other non-player-controlled creatures with similar properties. Plus, we found that the creature controlled by the player had similar properties in terms of vital energy and interactive abilities. From the analysis of the two concepts in the alluded and other reviewed games, we concluded that the category entity was proper enough to encompass all the biological and non-biological, player- and non-player-controlled agents endowed with some ability to interact with the environment, and having vital resources that can possibly be affected by specific interactions, eventually leading to the "death" of the agent.

As a second example, consider the concepts friendly and enemy. The observation of the playing session revealed that the player used the notion friendly to refer to all those agents that helped the entity that he controlled, whereas enemy was used to refer to all those entities whose purpose in the game was to harm the player-controlled entity. While analyzing other games, we found that the concepts enemy and friendly regarded also interactions amongst non-player-controlled entities, and concluded that attitude was a proper category to encompass all the concepts that define the disposition of an entity toward another.

Finally, consider the concept *Nali*. In the extract N. 2 the player used this notion to talk about who a specific entity was, i.e. he was talking about the entity's identity. Therefore, we concluded that the category identity was proper to encompass all those concepts that specify the condition or character as to who or what an entity is.

The former conceptualization examples are summarized in Table 3.

Table 3: Categorization of concepts

Concept(s)	Properties	Derived Category
Car, guy	- Interacts with the environment - Is “alive” and can “die” - Player- or non-player controlled - Biological- or non-biological	Entity
Friendly, enemy	- Define disposition of an entity toward another	Attitude
Nali	- Specifies the condition or character as to who or what an entity is	Identity

2.3.3 Description of Categories

Goal

Provide a description of all the categories found in the previous stage, covering all their relevant properties and how these are usually implemented in action videogames.

Actions

Formulate a description of the category using information provided by the players, and deducing additional information from the analysis of the implementation of the category in the sampled videogames.

For the descriptions of the categories, refer to section 3.

2.3.4 Analysis of Relationships between Categories

Goal

Identify relationships between different categories of concepts.

Actions

1) Interpret players’ explicit and implicit statements of relationships.
2) Derive a relational representation from the former analysis.

Guiding questions

- Do players explicitly relate this category to other ones?
- Do players speak about this category by referring to other ones?

Example

In this example we consider the sentence “He was a *Nali*, and they are friendly…” from extract N.2. In the first part of the sentence, the player talked about the <u>identity</u> of an <u>entity</u>, while in the second part he made an assertion regarding the <u>attitude</u> of those particular <u>entities</u>. This leads to the analysis in Table 4.

Table 4: Analysis of relationships between categories

Source	Data	Interpretation	Relationship
Extract N. 2	“He was a *Nali*…”	Player is speaking about an entity, saying who he is.	<u>Entities</u> have an <u>Identity</u>
Extract N. 2	“…they are friendly…”	Player is speaking about a group of entities, saying that they are friendly. Player relates friendliness with the identity of the group of entities.	<u>Identity</u> is related to <u>Attitude</u>

The former analysis leads to the following relational representation:

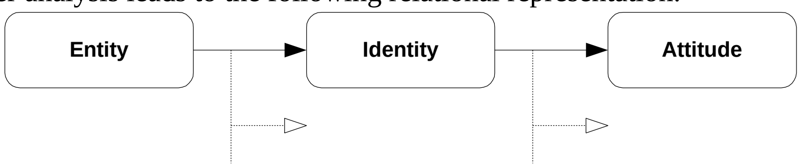

Figure 1: Hierarchical relationships

2.3.5 Analysis of Players' Preferences

Goal

Identify preferences expressed by players regarding each category of concepts, and deduce game design guidelines to improve the playability of a game.

Actions

1) For each category, analyze players' remarks regarding playability.
2) For each remark, understand whether it originates from design prescriptions or recommendations (if not explicitly stated, keywords like "always", "never", "mandatorily", can help to interpret the data). If it's not clear, ask players explicitly.

Guiding questions

- What did players say about this category?
- How did they relate it to playability?

Example

As shown in Table 5, in this example we consider a judgment made by the player regarding an identity-related problem that prevented him from identifying an entity in the game. Such a judgment leads to the deduction of a guideline to avoid it. Additionally, the presence of the word *always* in the player's statement indicates the mandatory character of the guideline, which must be therefore considered a prescription.

Table 5: Deduction of design guidelines

Source	Category	Data	Design guideline
Extract N. 2	Identity	"You should always be able to recognize your friends…"	Prescription: Allow the player to easily understand whether an entity is friendly, hostile or neutral to the player's token.

3 RESULTS: A QUALITATIVE GAME DESIGN REFERENCE

The analysis of the qualitative information gathered during the playing sessions led to the formulation of a hierarchical structure of categories of concepts considered important by players when judging the playability of an action videogame (Figure 2).

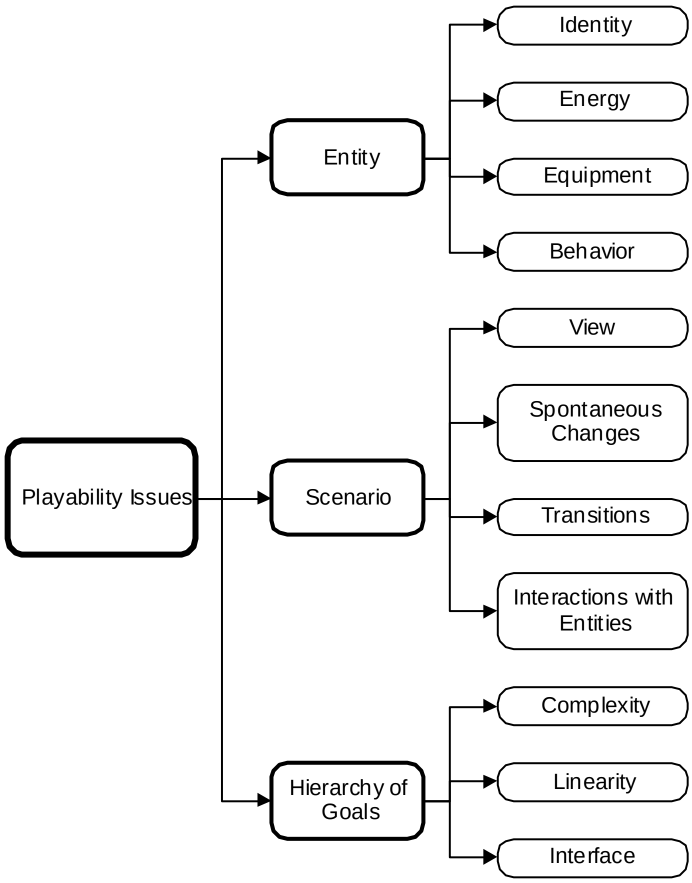

Figure 2: Overview of determinants of the playability in action videogames

Such concepts correspond to key aspects of a game design, and are related to specific guidelines derived from the analysis of players' preferences and focused on the improvement of the playability of action games. This makes the structure of concepts and guidelines useful as a game design reference.

The remainder of this section describes the structure and the related guidelines.

3.1 Entities

In action videogames the player is allowed to interact with a virtual world by controlling a token, which is the protagonist in the game and, consequently, the most important entity that inhabits the world. Additionally, the world is also populated by non-player-controlled entities. These can be antagonists (if they interfere with the protagonist's progress toward the game's goals) or variable-attitude entities, whose reactions towards the player's token may be friendly, hostile or indifferent, depending on the situation and the behavior of the token.

Entities are the first element upon which players focus their attention when judging a videogame, and the design aspects that determine the characteristics of an entity in terms of playability are shown in Figure 3.

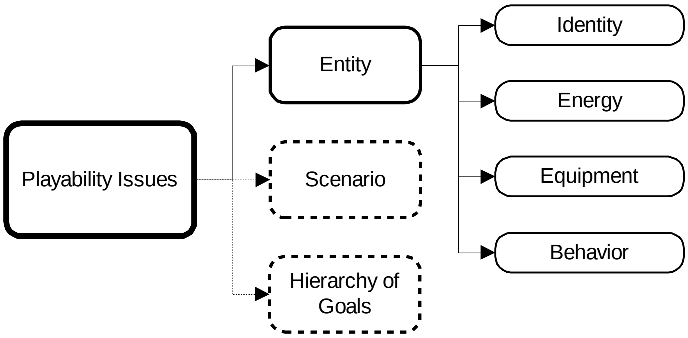

Figure 3: Design categories for a game entity

3.1.1 Identity

In order to interact with entities (both directly and indirectly), players must be able to gather and elaborate information necessary to decide what to do with them in specific situations. Such decisions depend on the perception of the entities' identity, defined by their role and attitude. Design aspects inherent to the identity of an entity are illustrated inFigure 4.

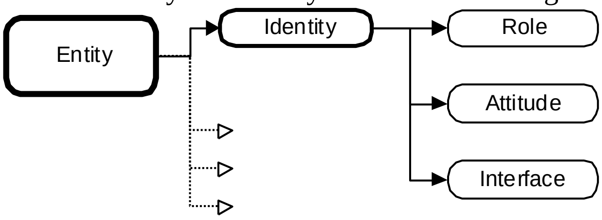

Figure 4: Design aspects of a character's identity

3.1.1.1 Role

In action games, the role of entities determines their abilities and purposes in the gaming world. Therefore, understanding the role allows the player to forecast entities' abilities (as in Eidos' `Thief: the Dark Project` where guards shouldn't be expected to have lock-picking skills – and in fact they don't) and attitude toward the player's token or other entities (for instance, guards in `Thief` shouldn't be friendly to the protagonist of the game, since he is a thief), even though this latter aspect could change during the game (see 3.1.1.2).

Additionally, some games allow players to choose the token's role, thus determining its specific abilities (as in Fox Interactive's `Alien Versus Predator`, where the player can choose to play as an *Alien*, a *Predator* or a *Colonial Marine*, thus determining the characteristics of the protagonist in terms of mobility, endurance, vision and cloaking, and the items and power-ups he/she will be able to find during the game-playing). Regarding the role, players' opinions led to the following design guidelines:

Design prescriptions

- Allow the player to understand the role easily, in order to make assumptions regarding entities' abilities and attitude. An exception can be made when the context of the game requires some exploration to understand the role (for example in the case of an undercover agent).
- The abilities of an entity should be coherent with its role. An exception can be made with games wherein fictional contexts make it impossible to have expectations regarding non-realistic characters.

Design recommendation

- Depending on the context of the game, offer the player the possibility of selecting the initial role of the protagonist of the game, to better suit his/her playing style.

3.1.1.2 Attitude

The attitude of non-player-controlled entities deeply influences their interactions with the protagonist and other entities. Therefore, the player needs to understand entities' attitude in order to make decisions to avoid wrong courses of action (as with *Nali* aliens in GT Interactive's `Unreal`: they are precious helpers, but their four-armed humanoid looks can make the player mistake them for enemies if he/she doesn't notice their non-hostile attitude). Design guidelines regarding entities' attitude are:

Design prescription

- Allow the player to easily understand whether an entity is friendly, hostile or neutral to the player's token. An exception can be made when the context makes discovering the attitude part of the challenge of the game, provided that there are no absurd ambiguities.

Design recommendation

- It is good to allow changes in entities' attitude to happen during the game-playing, provided that enough information is transmitted to avoid confusing the player.

3.1.1.3 Identity interface

Information about entities' role and attitude is transmitted to the player by means of an interface that has a contextual and an explicit component, as shown in Figure 10.

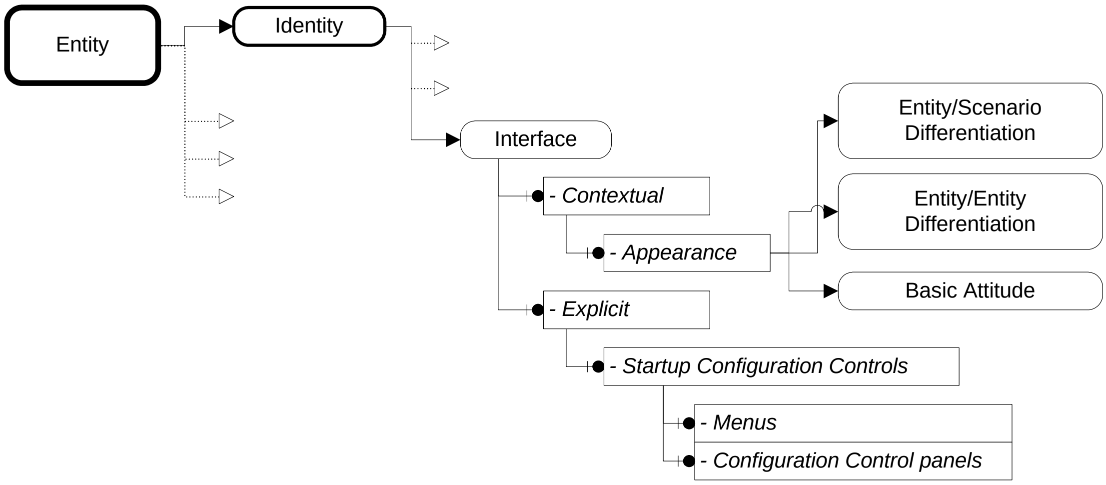

Figure 5: Identity interface

Contextual means

The contextual component uses the entity's appearance as a means to transmit information about its identity, often employing sound as a complement. Appearance is necessary to allow the player to identify entities and understand at least their basic attitude (for example, the uniform of a soldier can mark his being a hostile marine as in Sierra's `Half-Life`). Design guidelines regarding the contextual interface are:

Design prescriptions

- The appearance of an entity should always allow the player to identify it among others, or other elements of the scenario, especially when it comes to the player's token. An exception can be made when the entities have little or no relevance in terms of game-playing.
- Whenever using sound in order to transmit non-visual information regarding an entity's identity, ensure minimum aural differentiation among entities belonging to different genres.

Design recommendation

- Exploit the usage of sound to transmit information about entities' identity in order to allow the player to elaborate better strategies before making visual contact with the entities (especially important when dealing with antagonists).

Explicit means

The explicit component of the interface is primarily used as an input means, to set up the initial configuration of the protagonist's identity through the use of menus and configuration control panels (such as those that allow the player to select the protagonist's role in Activision's `Hexen II`). This led to the design guideline shown below:

Design prescription

- The explicit means used to configure the role of the player's token should be intuitive and easy to use, in order to avoid time-consuming customization processes.

3.1.2 Energy

The energy of an entity is a resource that allows its existence in the gaming world. Every entity has an initial energetic status, which can change as a result of an interaction with some other entity (such as an opponent) or element of the gaming world. Design aspects inherent to the energy of an entity are illustrated in Figure 6.

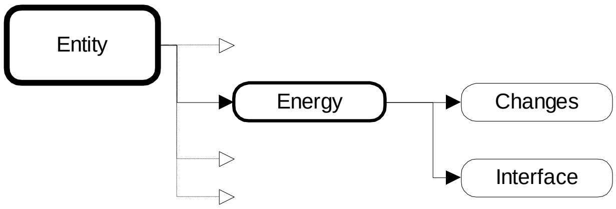

Figure 6: Design aspects of an entity's energy

3.1.2.1 Changes of energy

As shown in Figure 7, changes in the energetic status of an entity are normally due to losses and increases.

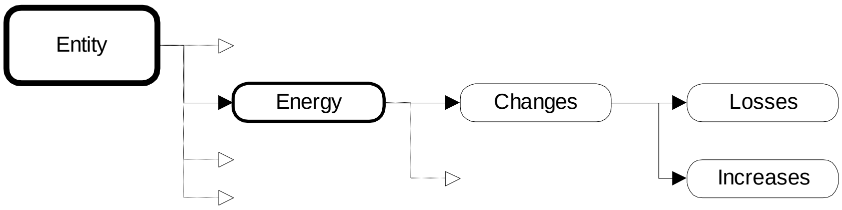

Figure 7: Energy changes

Entities lose energy as a result of an unfavorable conflict with other entities (as in Id's `Quake` combats) or an interaction with other harmful elements of the gaming world (like the spikes that cover the ground of some areas in Eidos' `Tomb raider III`).

In several games, the protagonist may reverse energy losses by interacting with healing elements (such as medikits in `Unreal`, or healing capsules in Origin's `Crusader: No Remorse`), some of which can even increase the amount of energy beyond the normal maximum (such as Acclaim's `Turok 2` *ultra-health* items, which always add 100 energy points to the protagonist's current status, whatever that may be). Players' opinions regarding changes of energy led to the following guidelines:

Design prescriptions

- The rationale that regulates energy losses should always be easily understandable.
- Energy increases should be caused by both items used as soon as the player's token picks them up (useful in fast-paced situations), and items that can be stored in an inventory for a later use.
- Restoration items should be located in the scenario according to the difficulty of different areas, and more powerful restoration items should be harder to find.
- Accessing and using restoration items should not be too dangerous, or at least not always dangerous.

Design recommendations

- It is recommended to provide precision-damage systems, which allow suffering or inflicting damages of different intensity depending on the parts affected.
- It is recommended to allow the player's token to completely restore its energy when moving to a new scenario or difficulty level, since it reduces the possibility of frustration of losing the game right at the beginning of something new.

3.1.2.2 Energy interface

Information about the energy status and changes is transmitted to the player by means of an interface with a contextual and an explicit component, as illustrated in Figure 8.

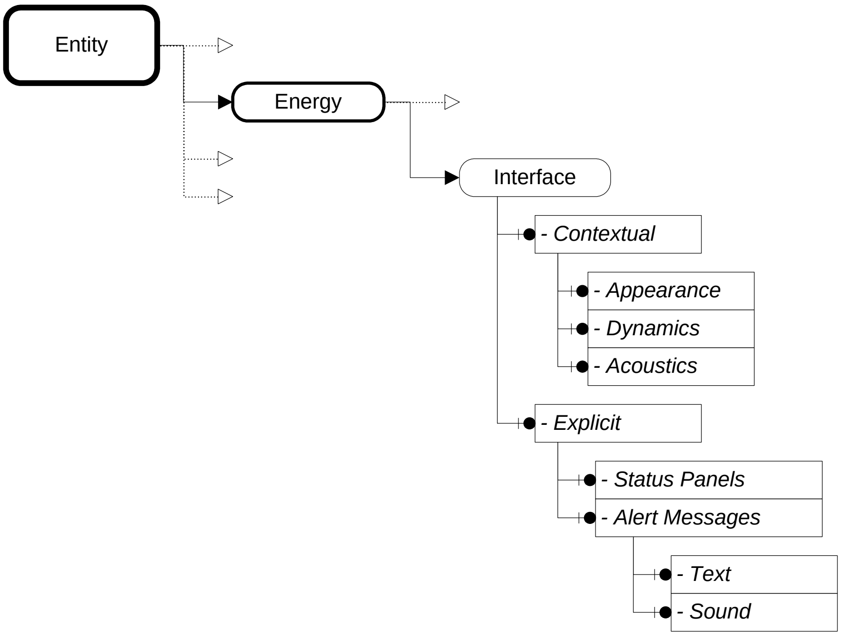

Figure 8: Energy interface

Contextual means

Contextual means can transmit qualitative information about a loss or an increase through details of the appearance of the entity (as the bruises and the smoke of a damaged car in BMG Interactive's `Grand Theft Auto`), the dynamics of its movement (as the swaying of a dying *Raptoid* in `Turok 2`), or the sound associated with events (as the exclamations of relief that accompany the use of a medikit in 3D Realms' `Duke Nukem 3D`). Design guidelines regarding the energy contextual interface are:

Design prescriptions

- Due to their immediacy, contextual means (especially entities' appearance) should be used to report energy losses whenever possible.
- Allow the player to have a precise perception of the magnitude of energy changes and the energetic status of relevant entities.
- Whenever using aural means to transmit non-visual information regarding an entity's energy, ensure that important sounds are not confused with or overwhelmed by less relevant ones.

Explicit means

The explicit component of the interface, implemented by means of status panels and acoustic or text-based alert messages, is normally used to transmit quantitative information about entities' energy in a very precise manner (as it happens with the protagonist's status bar that pictures available lives and energy points for the current life in Activision's `Earthworm Jim`). Players' opinions regarding explicit energy interfaces led to the following guidelines:

Design prescriptions

- Provide the player with at least some precise basic information about his/her token's health.
- Use explicit means whenever distance, size and/or some other structural characteristic make contextual means (such as the appearance) insufficient for the player to understand information about entities' health.
- Explicit controls used to display information about entities' energetic status should be understandable from the very beginning of the gaming experience, since trial and error may imply losing the game too often, with consequent frustration for the player.
- Explicit interface controls shouldn't clutter too much visual area.

Design recommendation

- When the game requires many explicit controls, it is useful to allow the player to choose which ones to display, and how to distribute them on the screen to better suit his/her needs.

3.1.3 Equipment

Entities often interact with the environment (including other entities) through the use of equipment. For a specific entity, the equipment can be defined as the set of items it can carry and use during the game-playing (for instance weapons, medikits, etc.). Design aspects related to the equipment of entities are illustrated in Figure 9.

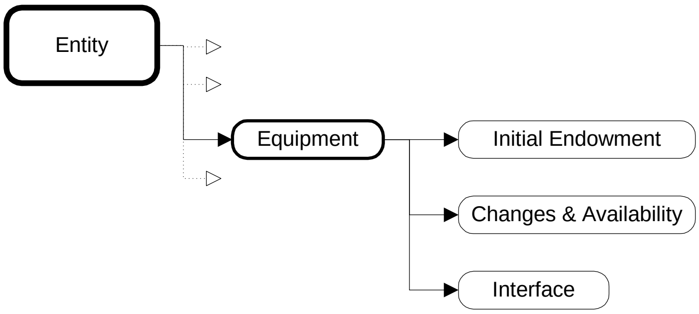

Figure 9: Design aspects of an entity's equipment

3.1.3.1 Initial endowment

Entities that can interact with equipment may begin the game with an initial endowment. In action games the initial endowment has different characteristics, depending on the type of entity it belongs to. Antagonists' initial equipment usually consists of one or more weapons (such as *Hunters*' blades and laser guns in `Unreal`) and/or other items (such as gangsters' cars in Activision's `Interstate '76`) that they can use against their foes. Variable-attitude entities and the player's token may have weapons as well as other items (such as flashlights, medikits, etc.), depending on the context of the game. Normally, the player's token begins the game with nothing more than a very basic weapon, and often such a weapon has an unlimited load of ammunition (such as the pistol in `Crusader: No Remorse`).

In racing games (such as `Need For Speed II` and Delphine's `Motoracer`) players consider the endowment issue in a slightly different way. Since normally each entity has nothing but the vehicle it is going to drive, the vehicle and the entity are considered as one. Therefore, the initial endowment can be constituted by enhancements that alter the basic performance of the vehicle (whenever such customizations are allowed). Guidelines regarding equipment initial endowment are:

Design prescription

- The player's token should never find itself unarmed and with no possibility of changing the situation for a long time. However, there should not be many unlimited weapons, and their availability should be determined according to the level of difficulty of the game, to avoid compromising the challenge.

Design recommendations

- It is of great help for beginners to start the game counting on a basic weapon with unlimited or very easy to find ammunition.
- It is good to allow the player to customize the initial endowment of equipment, in order to better meet his/her preferences and playing style. However, customization processes should not be too time-consuming.

3.1.3.2 Changes and availability of equipment

The initial status of entities' equipment can change as new items are acquired during the gameplaying. How the equipment of antagonists and variable-attitude entities changes is usually not evident to the player: from a given moment, the player's token simply faces better-equipped friends or foes. Regarding the token, the player determines changes in its equipment by collecting items (usually stored in an inventory), and dropping or using them. Therefore, such changes depend on the availability of items in the gaming world. Design guidelines regarding availability and changes of equipment are:

Design prescriptions

- The player's token should never be left without fundamental items for too long. Otherwise, the progress in the game could be compromised, with onset of frustration.
- The availability of the items (in terms of location and quantity) should be designed according to the situation that must be faced using them and depending on their effectiveness.
- The evolution of the endowment of equipment should not be a prerogative of the player's token.
- Whenever there are restrictions on the storage of items in the inventory, there should be mechanisms warning the player when the maximum allowed for a specific genre of items has been reached.

Design recommendations

- Counting on different items (especially weapons) that have different modes of use and effects enhances the gameplay.
- Every piece of equipment should have unique characteristics, to compel the player to adopt ad-hoc strategies.
- Distinctive characteristics should make items belonging to the same genre (for instance weapons) more or less effective, depending on the situation the player is facing.

3.1.3.3 Equipment interface

Information about entities' equipment is transmitted to the player by means of an interface with contextual and explicit components, as illustrated in Figure 10.

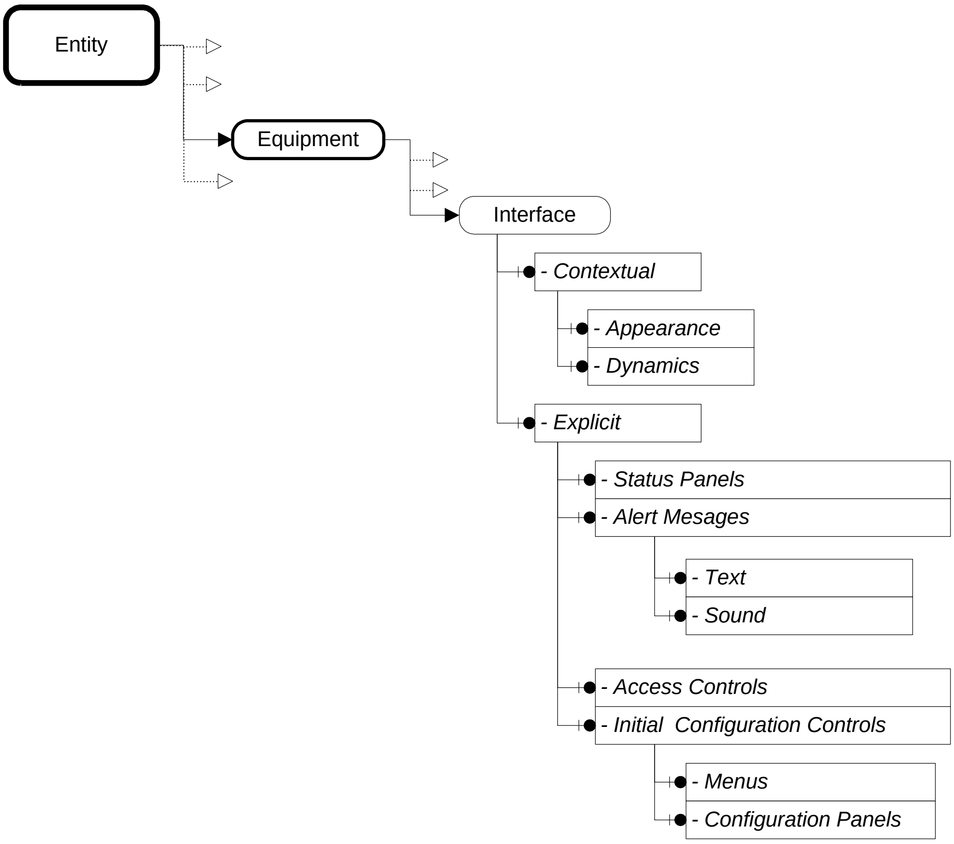

Figure 10: Equipment interface

Contextual means

To transmit information, the contextual component employs details of the appearance of the entity (such as the representation of the protagonist's hand gripping the weapon in use, in `Turok 2`) or its dynamics (such as a slow movement due to heavy equipment).

Design guidelines for the equipment contextual interface are:

Design prescriptions

- Appearance should always transmit some information about what an entity is wearing or using.
- The appearance of an item should transmit enough information about its semantics, potential and differences with other items belonging to the same genre (for instance weapons).

- Use explicit alert messages whenever the appearance of an item is not enough to transmit information about its semantics.
- Explicit representations of the in-use items can serve as a reference to perceive distances and proportions among objects, to find one's bearing and aim at a target (the latter especially in third-person view – see section 3.2.1.1.2).
- The rendering of in-use items should not occupy too much space in the visual area; otherwise, environmental visibility would suffer.
- Use entities' movements to transmit information about the status and availability of equipment.

Design recommendations

- Special-purpose items can be used to vary the gameplay (as in the case of remote-controlled weapons).
- In first-person view (see section 3.2.1.1.1), provide the option for a "hidden-weapon" mode, in order to allow the player to have a better visibility of the playing area.

Explicit means

The explicit component of the interface is employed only dealing with the player's token, with both input and output functionalities.

As an input means, it uses configuration controls (menus and panels) to: allow the setting of some initial properties of the token (a common practice in racing games, which often allow vehicle customization prior to the race); access items available in the inventory (as the weapon-selection or medikit controls in `Half-Life`); or control the aiming system. This can be: automatic, which does it all (the player simply has to pull the trigger to hit the target); semi-automatic, which requires the player to aim at the area near the target to hit it; and manual, which forces the player to precisely aim at the target.

As an output means, the explicit component of the equipment interface employs status panels to transmit information about the availability and status of both stored and in-use inventory items, and sound and/or text-based alert messages to provide explicit warnings when the player's token picks up or drops items. Additionally, explicit output means can also be used to implement visual aiming references (such as laser beams). Design guidelines for the equipment explicit interface are:

Design prescriptions

- Use explicit alert messages whenever the appearance of an item is not enough to transmit information about its semantics.
- During fast-paced action sequences, equipment access controls should be very quickly usable and straightforward, hopefully with one-to-one (one control for each item) or cycle-through controls (controls to cycle through a list of available items).
- Whenever a cycle-through option is available, players should be able to count on means to know which one is the next or previous item in the list.
- Whenever controlling an item prevents the control of the player's token, the situation should expose the token to dangers the player could not possibly react to.
- Aiming systems must be simple and immediately understandable, since misusing them could mean losing the game, with consequent onset of frustration.
- Aiming systems should be designed according to the situation the player is going to face during the game-playing.
- Manual aiming systems are more challenging than (semi) automatic ones. However, they are poorly playable when the player has to face many opponents simultaneously, since the pace of the action could prevent him/her from aiming precisely.
- Automatic and semi-automatic aiming systems benefit beginners, and are generally useful when the field of view does not allow aiming with accuracy (as in third-person view).
- Explicit aiming references benefit every kind of aiming system, especially when there are no contextual references (such as the explicit representation of part of a weapon in first-person). In such cases, there should be at least the option of having an explicit reference.
- Status panels should always be used to inform about the availability of items, with the same precision for all the elements that can be stored in the inventory (not knowing that certain items are available in the inventory may lead the player to make wrong decisions). Regarding other properties of items, the maximum level of details should be provided at least for the ones in use.
- Status panels should transmit information without compromising the size of the visible area.

- Text or voice-based alert messages indicating that the token has just picked up an item should be short and immediately understandable. This is possible when the items' names are carefully selected to evoke the semantics of the item.

Design recommendations

- It is useful to organize the most frequently accessed items to make their use easier, in order to allow players to quickly select the ones they consider most suitable to their needs (for instance grouping weapons according to their basic characteristics, and ordered within each group according to their power, thus allowing the player a quick selection of a group-power combination).
- Entities with similar equipment and abilities should have similar aiming efficiency.
- It is recommended to allow the customization of the number and position of visible panels.

3.1.4 Behavior

In action games, entities have behaviors that determine what they can do, and when and how they do it. Main design aspects of the behavior of an entity are illustrated in Figure 11.

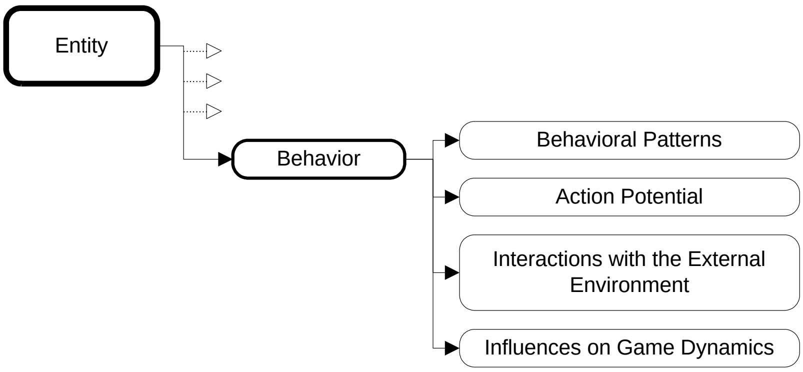

Figure 11: Design aspects of an entity's behavior

3.1.4.1 Behavioral patterns

Behavioral patterns are relevant only for non-player-controlled entities (since normally the protagonist acts according to the player's decisions), and are related to the aspects shown in Figure 12.

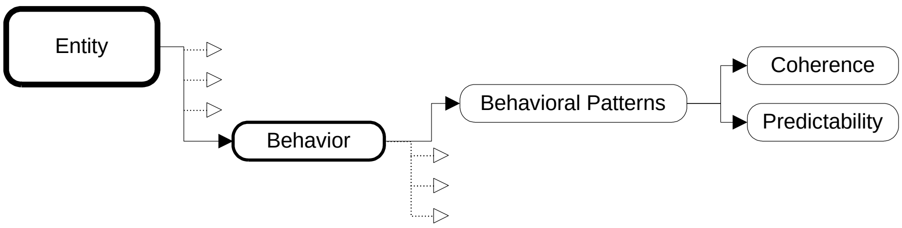

Figure 12: Behavioral patterns

3.1.4.1.1 Coherence

Coherence between an entity's behavior, role and abilities is a very important issue. An entity whose behavior does not seem to be logically related to its role, or suffers inexplicable changes, is said to have incoherent behavior (as in the case of `Half-Life`, where limited artificial intelligence in friendly security guards makes them expose themselves in an absurd way, without any care for their own health). Coherence could lead to predictable behaviors, although it's not necessarily so. In fact, in a given context entities could opt for different alternative behaviors, all perfectly coherent with the context. If the player had no prior knowledge of one of such behaviors, then this would be unpredictable. However, it would still be judged coherent at a later time. Design guidelines regarding behavior coherence are:

Design prescriptions

- Behavioral patterns should be coherent with the structural characteristics and potential of the entity and with whatever happens in the surroundings, changing whenever required by the situation.

- Complex behaviors should never seem confusing or random: there should always be a hint of logic in what each entity does.
- Changes in behavioral patterns must always be related to some reason evident to the player.

3.1.4.1.2 Predictability

In action videogames non-player-controlled entities act according to more or less non-deterministic pre-designed behavioral patterns, which determine what they will do given a set of external conditions.

Predictability as a criterion mainly pertains to antagonists. In simple designs (such as ghosts' behavior in `Pac-Man`) patterns are quite predictable, making it possible for the player to identify each one of them and the conditions under which they will be followed. In such cases, even very strong or well-equipped antagonists can become poor contenders. In more sophisticated games, patterns can be very complex and unpredictable, and players can easily be confused by unexpected elements. Unpredictability depends mainly on entities' intelligence, since intelligence is what determines their ability to decide what to do in a given circumstance. However, other intelligence-independent design elements (such as the location and number of certain variable-attitude entities, and how and when interactions with them can be triggered) can also be used to determine the unpredictability of some entities. All this leads to the following design guideline:

Design prescription

- Unpredictable behaviors enhance the challenge. However, it should always be possible to understand the reasons why things happen, and the patterns that regulate their happening, even if it takes some time.

3.1.4.2 Action-potential

The action-potential of an entity can be defined as the set of actions that the entity may perform. Such actions are combined in sequences and used to interact with the rest of the virtual world. Main design aspects of the action-potential are shown in Figure 13.

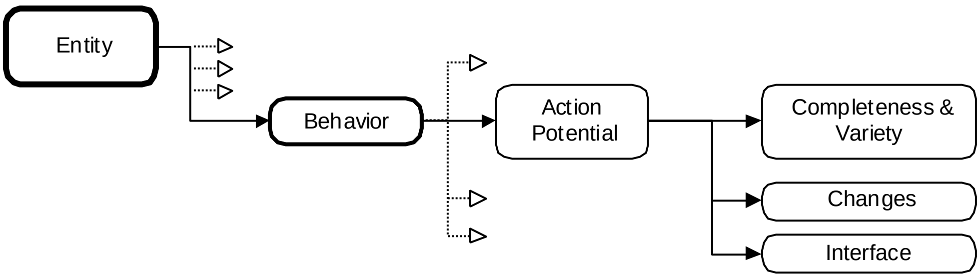

Figure 13: Design aspects of the action-potential

3.1.4.2.1 Completeness and variety

The action-potential (which includes fully, partially and non-player-controlled actions) is always very important to players, especially if it's the potential of the player's token. The potential is judged according to its completeness (i.e. how well the set of available actions is suitable to face all the possible gaming situations) and variety (i.e. how many alternatives of action are available to perform a specific task). Players consider a set of actions to be complete only if it allows them to do whatever is required by the game, in coherence with their expectations. Design guidelines regarding the completeness and variety of the player's token potential are:

Design prescriptions

- To ensure completeness, the potential should be designed according to the situations that the player is going to face during the game.
- In every situation it should be clear what actions might be needed and what would be useless. Available actions should be as few as possible, for the sake of simplicity.
- The degree of controllable actions should always be made very clear to players.
- If the potential includes non-player-controlled actions, it must be possible to understand their semantics and the reasons behind them.
- Dynamics of uncontrolled actions should be understandable to predict their effects on the token's status.

3.1.4.2.2 Changes

The action-potential of an entity can be static or dynamic.

In the case of a static action-potential, the actions performable by entities are always the same, even though they are not always available (for example, in `Pac-Man` the protagonist can devour enemy ghosts only for a while after eating a power pill).

Dynamic action-potential may change during the game-playing, and changes can be user-controlled (normally in the case of the player's token), or determined by events generated by the computer.

As for user-controlled changes, many games currently offer players the possibility of customizing some aspects of the potential of the player's token, or selecting from alternative tokens with different potential at the beginning of the game (as often happens in racing games like Electronic Arts' `Need for Speed II`). Additionally, there is often the possibility of changing the potential during the game-playing, in special customization stages, whose outcome usually depends on decisions made and tasks performed during previous stages of the game.

As for non-player-controlled changes, in some designs (like Sierra On-Line's `Half-Life`**),** as the game progresses, entities learn how to do new things, or acquire new capabilities thanks to items collected during the game-playing, all of which produces an evolution in the potential (as in `Turok 2`, where *Endtrails* begin level 1 with basic combat skills and improve them during the game-playing by means of better weapons and enhanced strategic abilities). In the case of non-player-controlled entities, such evolution is never evident: from a given moment, the player simply faces better opponents or counts on more skilled helpers and informers. In the case of the player's token, the evolution usually happens by means of the acquisition of new powers or equipment that improve current abilities and/or endow the token with new ones (as in `Hexen II`, where the protagonist finds more powerful weapons as he progresses in the game-playing).

Finally, a dynamic action-potential allows the player to perform the same tasks by means of different actions (as in `Unreal`, where the protagonist can often eliminate opponents using different weapons, or setting traps to avoid direct contact with them). Guidelines regarding the action-potential changes are:

Design prescriptions

- The context of the game must be coherent with the possibility of customizing the player's token at the beginning and even during the game-playing.
- Whenever present, customization phases must neither disrupt the game flow nor delay the beginning of the action.
- Startup or in-game action-potential customizations should never allow the creation of an invincible token (in order to prevent the game from becoming unchallenging).
- Changes in the token's potential should allow new actions to be performed, and not only to perform the same ones, but in a better way. This gives the player an opportunity to tackle the game following different strategic approaches and applying different gaming styles.

Design recommendations

- If customizations are allowed, offer computer-assisted options to minimize delays that might affect players interested in the action more than in any other aspect of the game.
- If startup configurations are allowed, offer a series of default alternatives to allow beginning the game as soon as possible.
- Having different genres of antagonists capable of performing different actions, thus requiring the player to vary his/her combat strategies, enhances the challenge of the game.

3.1.4.2.3 Action-potential interface

Both contextual and explicit interface means are used to transmit information about the action-potential, as shown in Figure 14.

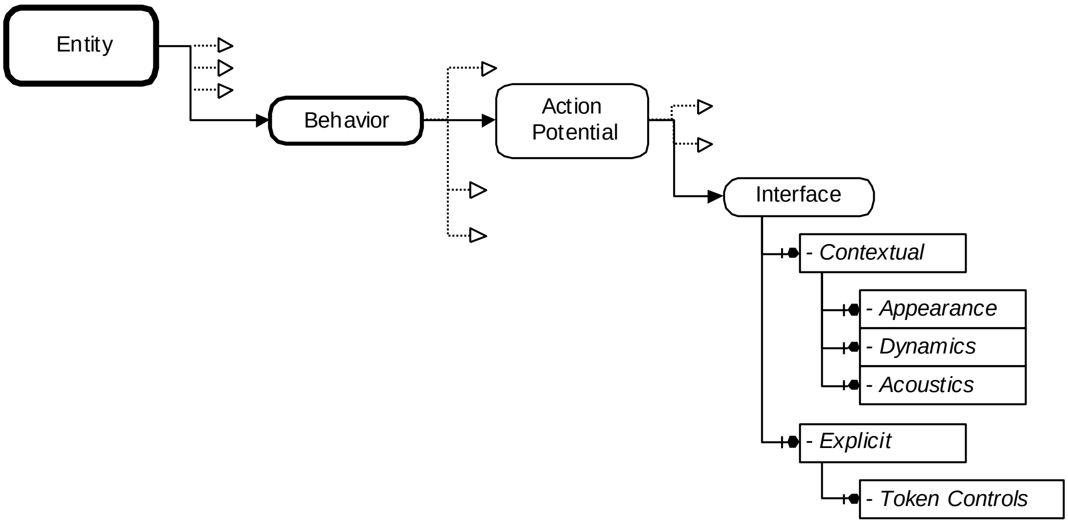

Figure 14: Action-potential interface

Contextual means

Aural and visual means, including entities' appearance, dynamics and acoustics, can be used to transmit contextualized information regarding the action-potential, according to the guidelines proposed below:

Design prescriptions

- Whenever possible, use the appearance of an entity to transmit contextual information about the action-potential, since it is the most immediately perceived means.
- Use aural means whenever possible, either because there is information that can be transmitted only through sound, or because visual information can be complemented by aural information.
- Detailed animations of entities' motion can help to transmit information about their potential. However, avoid non-player-controlled time-consuming actions that might disrupt the flow of the game.

Explicit means

Dealing with actions, the explicit component of the interface is used as an input means to allow the control of the player's token, and it is therefore a crucial element to exert control over the gameplay. Guidelines regarding the explicit interface are:

Design prescriptions

- Whenever possible avoid complex controls, relying on many keys, buttons and/or other devices, since they are very difficult to handle in fast-paced sequences. If that is not possible, ensure that at least the basic actions can be performed using a few, simple controls.
- If combinations of controls are needed to perform a specific action, the semantics of the action controlled by the combination must be logically related to the semantics of the actions controlled by the separate controls alone.
- Whenever combinations require the use of too many controls simultaneously, replace them with all-in-one custom controls (such as a "one button does it all" control).
- Let the player customize controls layout, and provide default layouts based on existing standards, in order to allow him/her to learn as few new things as possible when playing similar games.
- The player's token must be highly responsive, allowing players do whatever needed without unexpected and unexplainable time lags in the responses of the token. Eventually, the responsiveness of the token should be determined by the medium it moves in.

Design recommendation

- Using controls whose semantics changes according to the situation can reduce the overhead of controls.

3.1.4.3 Influences on game dynamics

Whatever the player's token does influences the game dynamics, since actions, interactions and their effects directly determine it. In action games, since action and a fast-paced game-flow are what really matters, time-consuming actions whose effects are not quickly evident (such as long dialogs with variable-attitude entities) are usually avoided, in order to eliminate potential sources of disruptions in the game-flow. All this leads to the following guideline:

Design prescription

- Avoid non-player-controlled feedback movements that could delay the responses of the player's token to player input (potentially disrupting the game-flow).

3.1.4.4 Interactions with the external environment

Game dynamics are mainly determined by how entities interact with the gaming world's environment, and by the consequences of such interactions. Possible interactions for a given entity are illustrated in Figure 15.

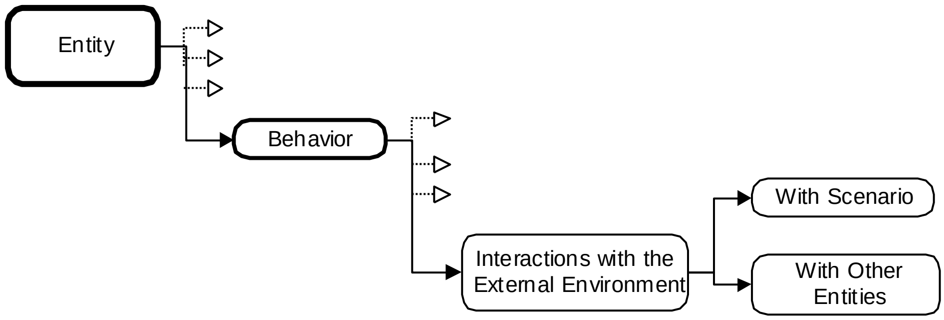

Figure 15: Interactions with the external environment

3.1.4.4.1 Interactions with the scenario

Interactions with the scenario are traditionally very limited in action games. More complex interactions are usually a prerogative of the player's token. Figure 16 illustrates the design aspects of the interactions between entities and the scenario.

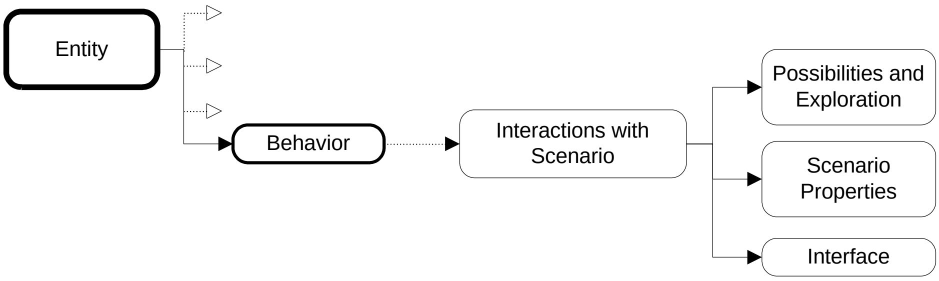

Figure 16: Design aspects of entity - scenario interactions

Possibilities and exploration

Depending on their consequences on the gaming world, interactions with the scenario can be classified according to two groups.

The first group includes all the interactions that have no relevant consequences (in terms of gameplay) on the status of the scenario, even though they could affect the participating entities. An example is the displacement of terrestrial entities. It is interaction with the scenario since it involves stepping on the ground; it usually has no significant consequence on the status of the ground; and it may have consequences on the displacement of the entity (such as in `Need for Speed II`, where the speed and responsiveness of the car changes according to the characteristics of the road). Objects involved in interactions belonging to this group can be defined as *inert*, since their status cannot be altered as a result of specific interactions.

The second group of interactions involves *active* objects, whose status is changed as a consequence of some interaction. These changes modify the status of the gaming world and affect the gameplay (as is the case in `Unreal`, where the protagonist can destroy walls to reveal secret passages, or move crates and explosive barrels to use them against the antagonists).

Exploration is a peculiar interactive possibility, often very important. In action games exploration is required to find things, information or places necessary to progress in the game.

Other common interactions involve scenario objects as strategic elements (for instance, in `Crusader: No Regret`, players often use explosive barrels and other objects to elaborate defensive and/or offensive tactics during the action), or as obstacles capable of interfering with the player's progress, possibly harming the player's token (as do broken pipes with gas leaks do in `Crusader: No Regret`). Design guidelines regarding exploration and other interactions with scenario elements are:

Design prescriptions
- Ensure a minimum level of interactivity: things that are expected to happen (according to the model of rules that regulate the game) should always happen.
- Due to the nature of the action genre (i.e. focus on action), minimize the set of possible interactions, including only the indispensable ones.
- Avoid time-consuming interactions with active elements, in order to prevent game-flow disruptions.
- Interactions with objects shouldn't require high psychomotor accuracy just to establish contact with the target object. This must be taken into account when determining objects' size, shape and location.
- Ensure that explorative processes do not disrupt the game-flow, and they don't replace action for too long. When exploration is mandatory, make it clear to the player.
- The type, number and location of inert and active scenario objects should be determined taking into account the fact that the player might want to use them as strategic elements in particular situations.
- The strategic use of scenario objects is not a prerogative of the player's token.
- Entities similar in terms of potential, equipment and structural characteristics should be able to perform similar types of interactions, and be intelligent enough to allow them to exploit their possibilities.

Design recommendation
- Automatically triggering interactions with objects by simply approaching them greatly benefits the game-flow, preventing disruptions.

Scenario properties

Interactions with the scenario depend on important properties of the elements that constitute it. Players pay attention mainly to the semantics of the scenario, space-time relationships and properties, interactive potential and cause-effect relationships among scenario objects, as shown in Figure 17.

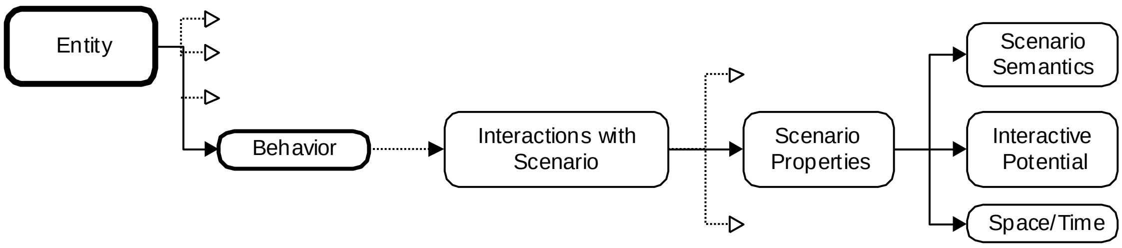

Figure 17: Scenario properties

Design guidelines regarding scenario properties are:

Design prescriptions
- The player must be able to clearly understand the semantics of objects in order to interact with them. If a learning process is required, it should not take too long, in order to avoid game-flow disruptions.
- Space-time relationships (e.g. distances between objects) and properties (e.g. the strength of the current of a river) should provide clues on how difficult interacting with certain objects could be.
- The possibilities of interacting with specific objects and/or changing their status, should always be evident. However, the purpose of interactions doesn't need to be clear at first glance (this can be part of the challenge).
- Similar objects should offer similar possibilities of interaction, with similar consequences on the status of the gaming world.
- Consequences of interactions should always be clearly perceivable and coherent with the physics of the environment. This meets players' expectations and allows some strategic planning to be done.
- Irreversible negative consequences of interactions must be avoided if they prevent the player from winning the game.

- In order to understand and possibly predict the results of the interactions, players must be able to understand cause-effect relationships amongst elements of the scenario. Trial-and-error is an acceptable method for this, provided that players never have to spend too much time trying to understand things.
- If chains of interactions are required to achieve a specific goal, in case of failure of intermediate steps the player must be able to tell what went wrong and when.

Design recommendation

- The player should be allowed to understand whether the token has already interacted with a specific object, in order to avoid unnecessary repetition of interactions.

Interface

In order to interact with the scenario, players need to receive important decisional information such as the semantics, possibilities and consequences of interactions, and the fundamental properties of the scenario. It is also necessary to provide information about the status of interactions and their consequences, to allow players to understand the relevance of interactions in terms of gameplay. Information about interactions between an entity and the scenario is transmitted to the player through an interface structured as shown in Figure 18.

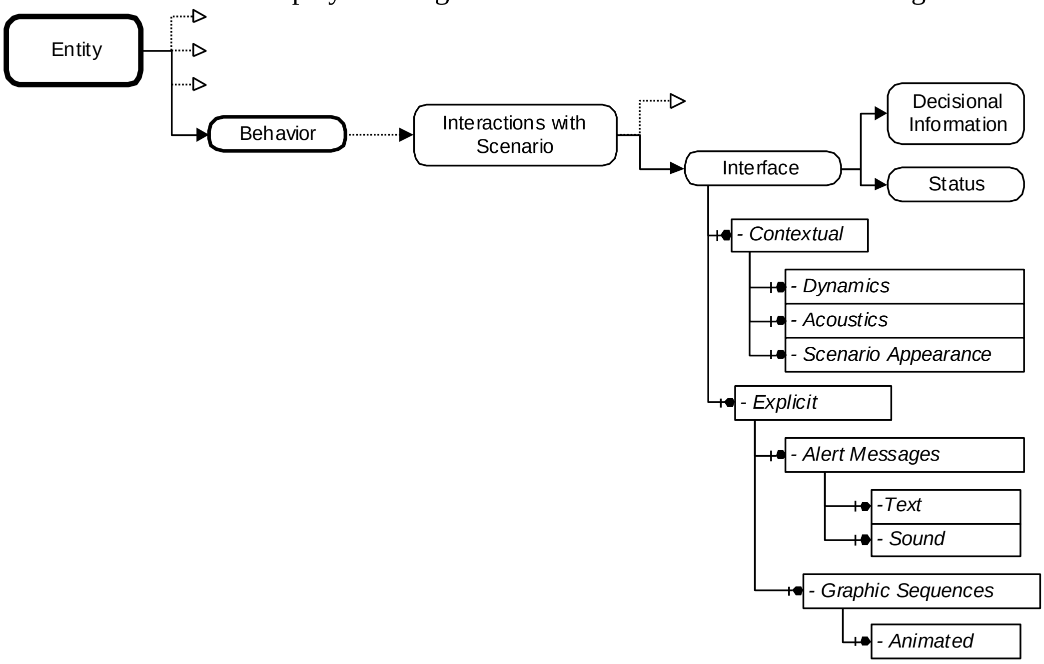

Figure 18: Entity - Scenario interaction interface

Contextual means

Due to its immediateness, the contextual component of the interface is the main means used to transmit information regarding entity-scenario interactions. Since during the gameplaying players are constantly looking at the scenario of the game, its rendering is an especially important means of transmitting information about semantics, space-time and cause-effect relationships, and about the possibilities of interaction with each object.

The quality of the scenario rendering and the players' viewpoint are crucial design issues. The former is mainly determined by the use of perspective, representations of space-time properties and relationships, and ambient lighting, while the latter depends on the position and viewing direction of the player's token (since the eyes of the token are what allows the player to watch the gaming world – see section 3.2.1). Design guidelines apropos contextual means are:

Design prescriptions

- A poorly rendered scenario may lead the player to misjudge the situation and make wrong decisions about where to go and what to do. The rendering must be clear and precise independently of the position of the player's token in the gaming world. Precision should ensure the visibility of relevant objects, depending on the position of the player's token.
- Visibility should not be illogically altered by specific actions performable by the token.

- The lighting of the scenario must always be coherent with the expectations of the player, and it should always ensure minimum visibility.
- The number and types of textures must be carefully selected. Too few textures could prevent a proper differentiation of objects, but too many of them could confuse the player, altering his/her perception of space-time relationships existing among objects or parts of a single object.
- The rendering should allow players to tell active from inert objects.
- Ensure the maximum possible visibility of those items that could harm the player's token (unless doing otherwise is part of the challenge).
- The dynamics of the motion of entities must change according to the characteristics of the part of the scenario they are interacting with and the structural characteristics and potential of the entity.
- Use environmental sound to transmit information about cause-effect relationships whenever the consequences of interactions occur within areas not visible to the player.
- In order to transmit semantics with precision, sounds must always be used in a realistic way, as the player would expect.

Design recommendation

- Textures and chromatic differences help in distinguishing objects and understanding some of the characteristics of the landscapes.

Explicit means

Explicit means are used in action games when dealing with possibilities or consequences of interactions. Text or sound-based alert messages are considered very useful when the player needs very specific information in order to interact with certain elements (as in LucasArts' `Dark Forces`, where key-cards of specific colors are needed to open certain doors, and text-based messages inform about the required card). Animated graphic sequences, used as cinematic cut-scenes (short movies automatically triggered as a result of interactions), are explicit means that can be effectively employed to transmit information about cause-effect relationships, especially the consequences of interactions. Design guidelines regarding explicit interface means used to transmit information about entity-scenario interactions are:

Design prescriptions

- Use sound and/or text-based alert messages to resolve ambiguities in the representation of the semantics of objects through their visual rendering.
- Cut-scenes should not tell everything about the consequences of interactions (a little exploration should always be a component of the challenge).
- The usage of cut-scenes should be limited to cases where the consequences of interactions are complex or distant from the token's location, and hence poorly visible.
- Cut-scenes should be kept short to prevent disruptions in the game-flow.

3.1.4.4.2 Interactions among entities

Interactions among entities draw players' attention during most of a gaming session. The design aspects of all the possible interactions for a given entity are shown in Figure 19.

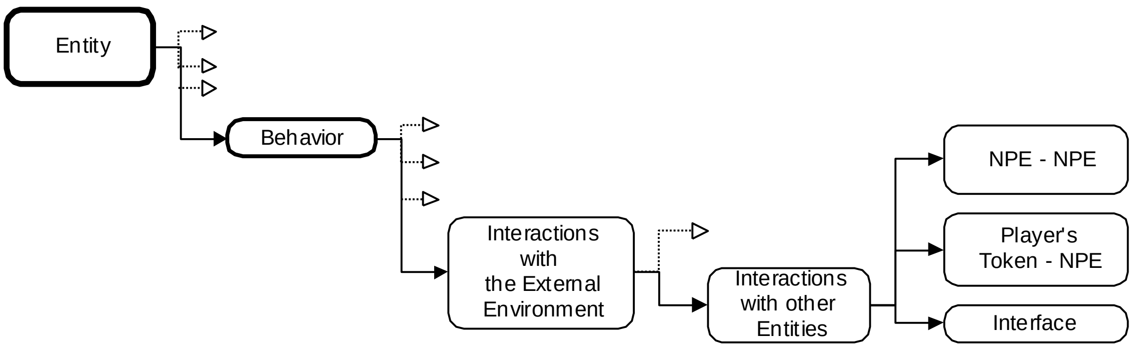

Figure 19: Design aspects of interactions among entities (NPE: non-player-controlled entity)

A given entity can interact with non-player-controlled entities or with the player's token. Considering that the entity could be the player's token, an antagonist or a variable-attitude entity, the resulting set of possible interactions is shown in Table 6.

Table 6: Possible interactions among entities (C marks interactions as possible and common in action games, while P marks them as possible but not common)

	Player's Token	Antagonist	Variable-attitude entity
Player's Token		C	C
Antagonist	C	C	P
Variable-attitude entity	C	P	P

General guidelines that apply to entity-entity interactions are shown in Table 7.

Table 7: Entity - entity interactions general design guidelines

Design prescriptions
Avoid useless or irrelevant interactions. Interactions must always be designed according to the main purpose of the game.
Similar interactions should have different outcomes, depending on the characteristics of the entities involved and other environmental conditions.
Whenever expected interactions are not possible, the player must be able to understand why.

Interactions between the player's token and a non-player-controlled entity

Such interactions are the main result of player's participation in the game-playing. Main design aspects of token - non-player-controlled entities interactions are illustrated in Figure 20.

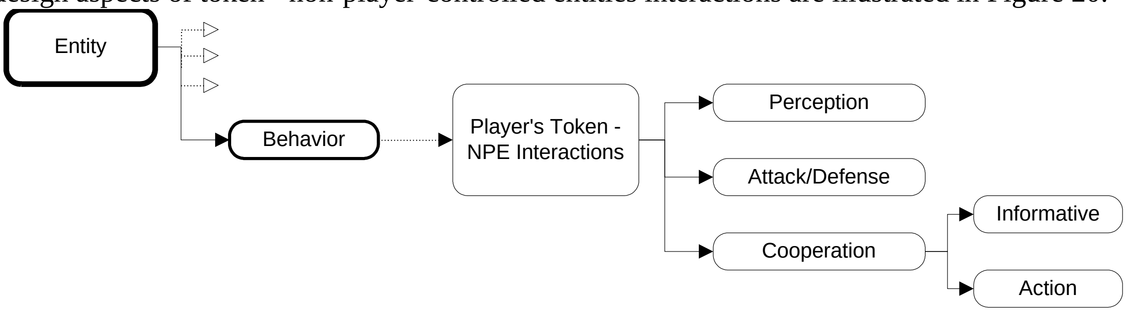

Figure 20: Design aspects of player's token - non-player-controlled entities interactions (NPE: non-player-controlled entity

The number and position of non-player-controlled entities are key elements for determining the possibility of interacting with them, which is particularly relevant when dealing with antagonists.

In order to interact with the player's token, non-player-controlled entities must perceive it. Sight-perception is the most common mechanism found in action games, and it allows non-player-controlled entities to perceive the token only when it enters their visual field. Other mechanisms are employed as well, such as aural perception and tactile perception (based on contact with body parts). Although initial perception is what allows an entity to interact with others, the development of interactions is from then on determined also by how the entity perceives what the others are doing, and by its ability to anticipate what the others are going to do. Nowadays sophisticated perceptual systems (like the one used in Shiny Entertainment's `MDK`) allow an entity to consider the complex characteristics of the motion of other entities (for instance, not only their position, but also their speed and direction), and make assumptions about their future status in order to decide how to interact with them.

Interactions between antagonists and the player's token are always offensive or defensive, and depend on the player's ability and the antagonist's offensive and defensive intelligence. In specific situations variable-attitude entities could also act aggressively, thus becoming antagonists, but it's rather unusual. In older designs, antagonists had a somewhat suicidal attitude: all they did was perceive the player's token and try to eliminate it as long as it was visible to them, with little concern for their own health (as in Id's `Doom`). Nowadays, many games make the player face AI-endowed entities that have different combat/competition strategies (as in the case of `Unreal`'s *Hunters* and `Turok 2`'s *Endtrails*), often based on the perception of the reality they are facing (for instance how heavily armed their opponents are), the equipment they are endowed with (to decide what is the best item to use in each situation) and their structural characteristics (as it is the case in Interplay's `Die by the Sword`, where *Cobolds*, who are pretty short, attack mainly their opponents' legs, without exposing themselves too much).

Interactions between variable-attitude entities and the players' token are usually cooperative, which enriches the gameplay. Variable-attitude entities collaborate with the player's token, providing information (e.g. scientists in `Half-Life`) or useful items and weapons (e.g. *Nali* people in `Unreal`). In sophisticated designs, variable-attitude entities also participate in the action (e.g. security guards in `Half-Life` and anti-terrorism squad members in Red Storm Entertainment's `Rainbow Six`), relying on their offensive/defensive intelligence and their equipment. Guidelines regarding player's token - non-player-controlled entity interactions are:

Design prescriptions

- Antagonists' positions must be sensible and their number balanced with their skills, strength and the situation of the game.
- A minimum level of interactivity should be allowed with every non-player-controlled entity, at least to make the player understand the entities' relevance in the game.
- Non-player-controlled entities' perceptual means must always ensure that the player's token will be perceived when it should be (for instance due to its position).
- Whenever used, tactile perception should always be implemented to ensure credible behavior (for instance, antagonists that are hit by a shot should react even if they cannot see their opponent, and have no aural perception).
- It should always be possible to notice whether antagonists have seen the token or not, and to understand where and when the token could be spotted, in order to elaborate strategies to take enemies by surprise or simply to protect the token.
- The efficiency of perceptual means should vary according to the situation (for instance due to the spatial properties of the perceiving entity - such as its position and orientation).
- Entities with similar characteristics in terms of structure, potential and equipment should also have similar perceptual capabilities/possibilities.
- Non-player-controlled entities' AI should allow them to show non-deterministic, non-suicidal behavior coherent with the situations they are facing.
- Under similar conditions in terms of skills, position, health and equipment, antagonists and the player's token should have similar possibilities of victory.
- Opponents that are expected to be smart should be smart, but poor offensive and/or defensive intelligence can be acceptable if the role of the entity justifies it.
- If a variable-attitude entity has something important to tell the protagonist of the game, informative interactions should be triggered automatically, as soon as the entity perceives the player's token. If the information is not crucial, it is reasonable to let its collection be the result of explorative processes based on interactions deliberately triggered by the player.
- Helpers should always act coherently with players' expectations.
- Instructions necessary to make helpers participate in the action (which must be neither too complex nor too many) should be imparted only once, and then remembered by cooperators.
- Assistants should not interfere with the course of action of the player's token.

Interactions among non-player-controlled entities

In many games entities have a life of their own, and don't live merely to face or interact with the protagonist of the game. This determines the importance of interactions among non-player-controlled entities (whose main aspects are shown in Figure 21) in terms of game design.

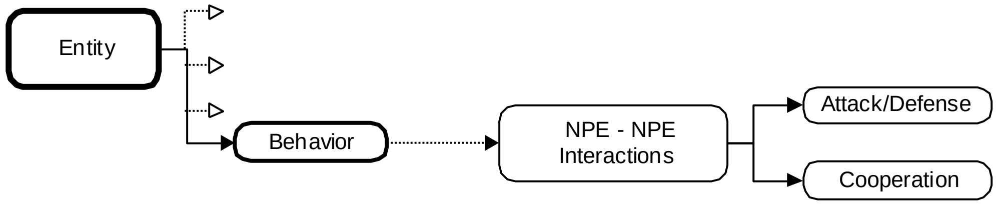

Figure 21: Design aspects of NPE - NPE interactions (NPE: non-player-controlled entity)

Interactions among non-player-controlled entities can be conflictive/competitive or cooperative. Interactions among antagonists are the most commonly found in action games.

Usually in action games antagonists can fight each other (which is the most common form of interaction, found even in old games like `Doom`), or can ally and cooperate against their foes (as in `Grand Theft Auto`, where police patrols collaborate when pursuing the protagonist). In both cases the complexity and efficacy of interactions depend on the intelligence of the antagonists, which determines their capacity to attack foes, defend or cooperate with others in collaborative tasks.

In some games, it is also possible to see antagonists interacting with variable-attitude entities. Sometimes variable-attitude entities are the victims of antagonists' attacks (as in `Unreal`, where antagonists often attack *Nali* aliens in order to prevent them from helping the protagonist). Otherwise they collaborate with the player's token against the antagonists (as the security guards in `Half-Life` do).

Usually variable-attitude entities don't interact with each other. Sometimes they belong to a party or a squadron, and collaborate with the protagonist in strategic missions (as in `Rainbow Six`), but nothing more complex than that is found in action games.

The guidelines regarding interactions between a non-player-controlled entity and the player's token (Figure 20) also apply to interactions among non-player-controlled entities.

Interface

Information about semantics and status of interactions among entities is transmitted to the player through an interface with an explicit and a contextual component, structured as shown in Figure 22.

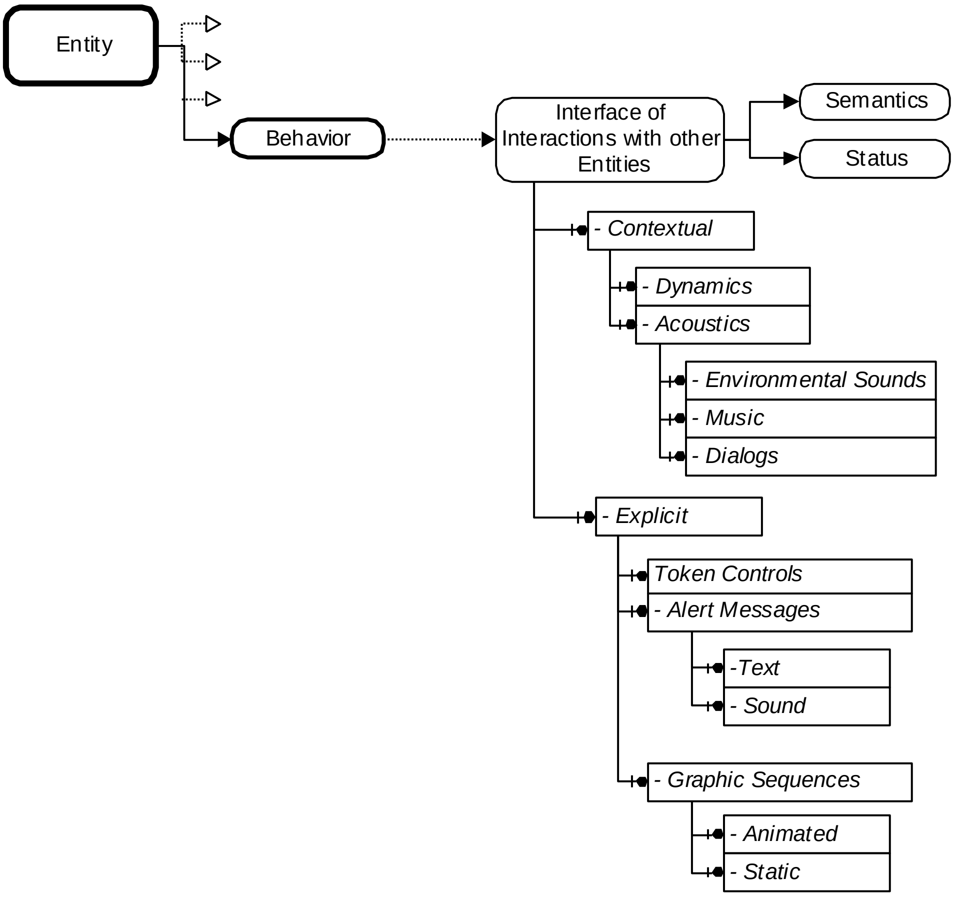

Figure 22: Interface of interactions with other entities

Contextual means

The contextual component of the interface uses aural means (both incidental sounds, associated to specific events, and environmental music, associated to specific situations in the game-playing) and the dynamics of entities' motion to transmit information about interactions (for example, in `Earthworm Jim` a cry, the flapping of feathers and the swaying of a hostile crow are all elements used to indicate that the bird was shot). Design guidelines regarding interface contextual means are:

Design prescriptions

- Use motion to transmit information about the consequences of interactions such as a fight, since it's a clear and immediately perceived means.
- Use incidental sounds to transmit not clearly visible results of certain interactions and information about the position of entities, or to reinforce information transmitted using other means.
- Use dramatic changes in the environmental music to indicate when relevant events occur.

Explicit means

In action games the explicit interface can be used to transmit precise information about the semantics of actions, using both text and sound-based alert messages or non user-controlled graphic sequences (such as cinematic cut-scenes or comics-like static sequences). However this is not at all frequent. Consequently, during the playing sessions players did not consider the explicit means as a relevant design issue.

3.2 Scenario

The second most important issue that determines players' perceived quality of an action videogame is the scenario, which is where the action takes place. Design aspects that define the scenario can be organized as shown in Figure 23.

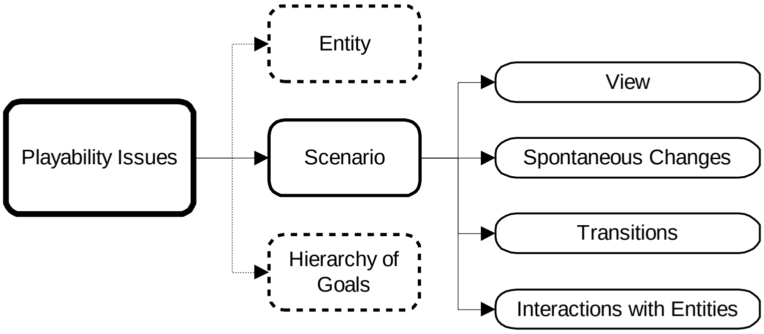

Figure 23: Design categories for the game scenario

3.2.1 Player's view

The player actually sees the scenario by means of a camera (a virtual eye) placed in a viewpoint with a given orientation (referred to as camera direction), and visibility is always limited to the viewing frustum, a pyramidal visual field whose axis is the camera direction (Figure 24).

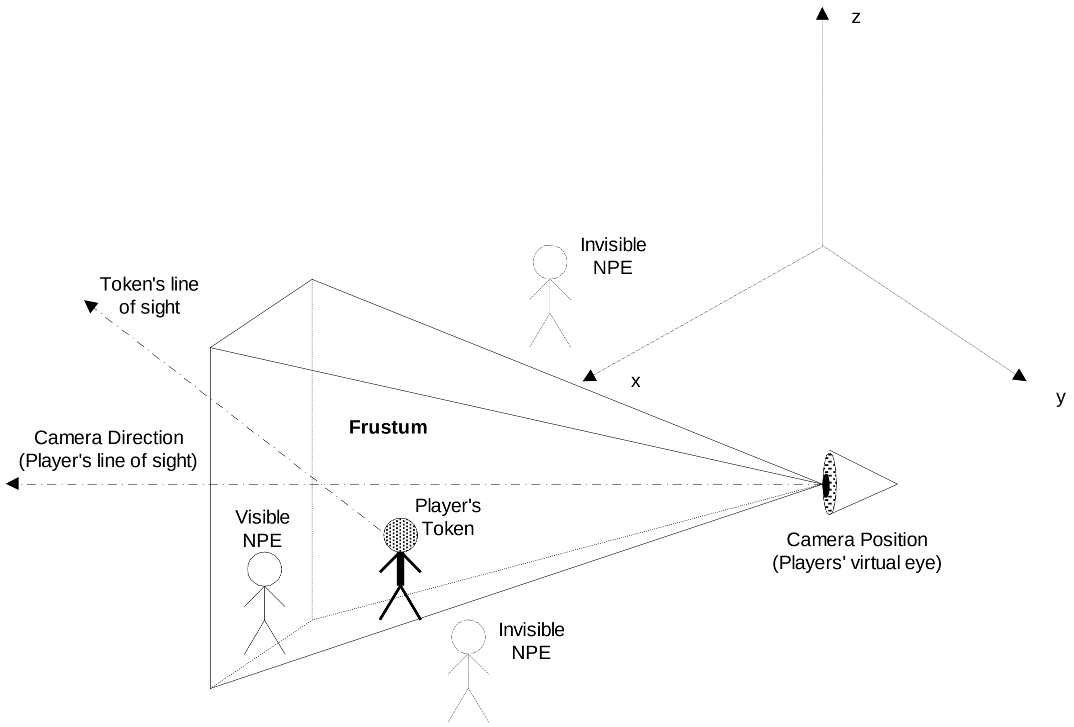

Figure 24: Player's view (NPE: non-player-controlled entity)

Therefore, how the player views the protagonist and his/her surroundings depends on the viewpoint and camera direction. Design aspects related to the view are shown in Figure 25.

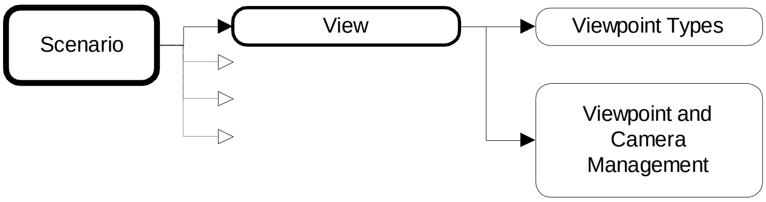

Figure 25: Design aspects of the player's view

3.2.1.1 Viewpoint types

As shown in Figure 26, action games use two different types of viewpoint: first-person and third-person.

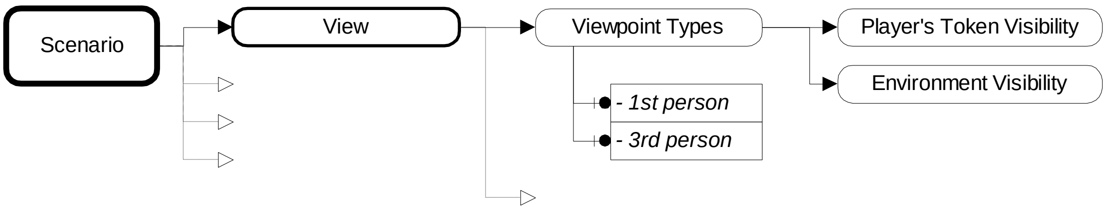

Figure 26: Design aspects of the viewpoint types

3.2.1.1.1 First-person view

In first-person view, the camera is located in the position of the player's token, and the player looks at the scenario as if he/she was the protagonist. Therefore, there is never a complete explicit representation of the player's token (since when you walk you can only see part of yourself) and environmental visibility is limited to whatever falls within the frustum of the token, as shown in Figure 27 (b). The field of view is usually very well handled, thus allowing a good perception of space-time relationships among scenario objects.

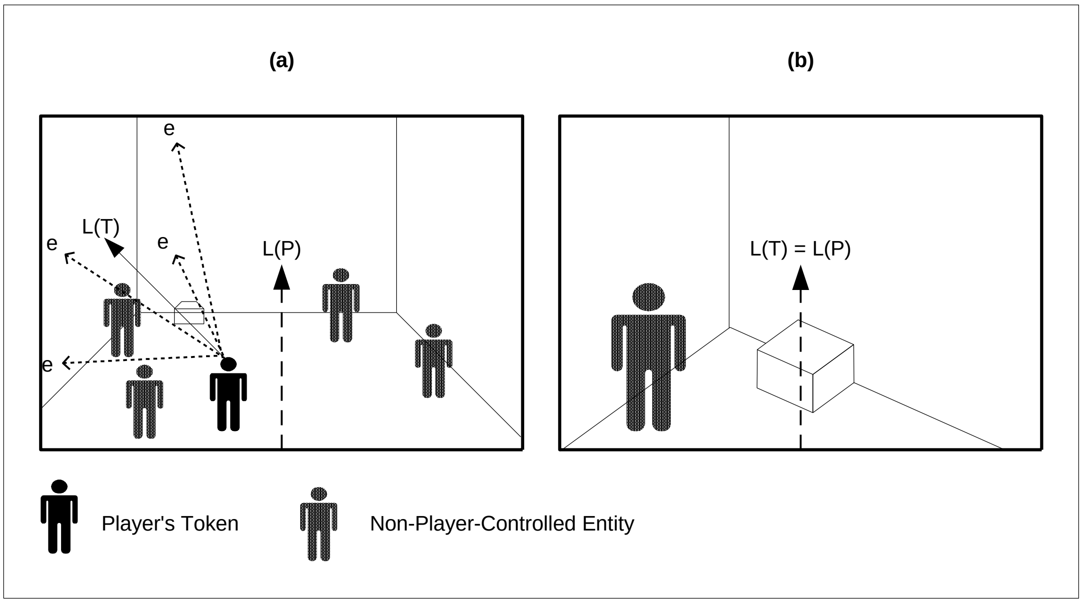

Figure 27: (a) Example of third-person view. (b) Same scene in first-person view (L(P): Player's line of sight; L(T) Player token's line of sight; e: edges of the token's frustum)

First-person may have drawbacks inherent to the degrees of freedom allowed to change the camera direction (and hence the frustum). For example, in games like `Doom`, which offers only four degrees of freedom in the movement of the frustum (determined by the X and Y axes, and the two possible directions for each axis), in order to see something located above the current position of the frustum, the player has to move the token to a higher position.

3.2.1.1.2 Third-person view

Third-person view (which is like a puppeteer's view) provides a complete representation of the protagonist (since the camera is not placed in the position of the player's token), and allows the player to see parts of the scenario that would remain hidden from his/her sight in the real world (such as the area behind the protagonist), as shown in Figure 27 (a).

Consequently, third-person provides a view of the scenario wider than the one provided by first-person, thus favoring strategic thinking (even if affecting the realism of the game). However, third-person can severely compromise playability when the player has to aim at targets with precision, since it does not provide a representation of the target as the player would see it if she/he were in the scene taking aim. Design guidelines regarding viewpoint types are:

Design prescriptions

- The visible area should be as large as possible, depending on the desired size of the virtual world elements (the bigger the size, the smaller the visible area).
- When the gaming area is not entirely visible, the scrolling of the scenario (i.e., movements of the camera and viewpoint to display new areas) must be activated before the token reaches the boundaries of the physical display. Otherwise, the player risks running into antagonists or obstacles without having the necessary time to react to their presence.
- When using a first-person view, be sure to provide three degrees of visual freedom (determined by the three coordinate axes, X, Y and Z) whenever it can be useful.
- When using a third-person view, minimize the occasions in which the player's token is not visible, and guarantee that nothing bad can happen when the player cannot control the token due to the lack of visibility.
- When using a third-person view, use the representation of the player's token (i.e., motion and appearance) to transmit information about what it is doing.
- Make visual aiming references (such as a cross-hair) available whenever precision aiming is needed while playing in third-person.

3.2.1.2 Viewpoint and camera management

As illustrated in Figure 28, viewpoints may change during the game, and, with third-person view, camera orientation and position may vary with respect to the protagonist's line of sight and position, thus providing different possible views for the same scene (for instance, in a third-person game it could be possible to view the protagonist from a direction aiming at his left side, right side or at his back, as in `Tomb Raider III`).

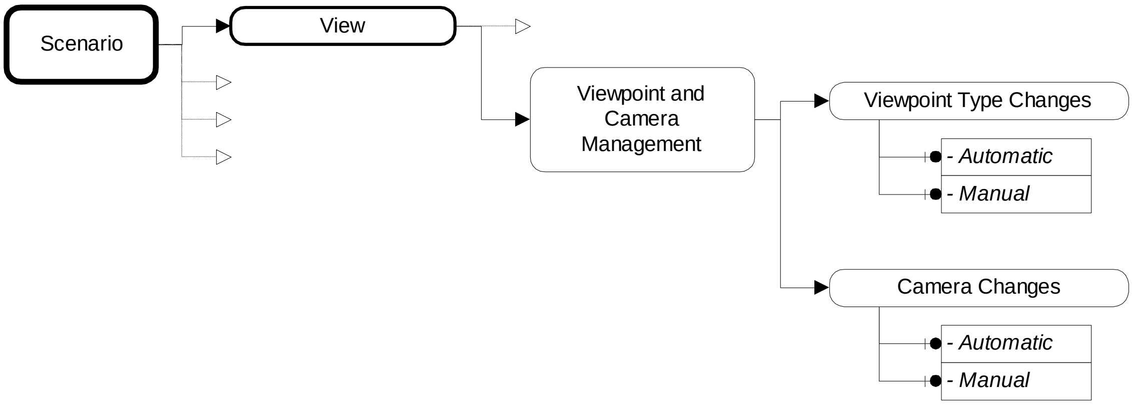

Figure 28: Design aspects of viewpoint and camera management

Even though many games adopt a specific viewpoint during the whole game, some allow swapping from first to third-person (as in the case of `Jedi Knight`). Such changes may be controlled by the player (as in `Jedi Knight`), or by the computer (i.e. automatically), depending on the characteristics of the ongoing action (for instance, Eidos' `Omikron: The Nomad Soul` uses a third-person view during the most of the game, but changes automatically to first-person when the protagonist drives a car). Changes in camera orientation and position may also be computer or player-controlled. The former are pre-designed, in order to give the best possible view of the player's token and its surroundings depending on the task it is performing (as in `Tomb Raider III`). The latter allow the player to choose the view that better suits his/her needs or gaming style (as in `Need For Speed II`, where the player can use a first-person view, looking at the track from the cockpit of the car, or a third-person view, looking at the track from the back or the front of the car). Design guidelines that apply to viewpoint and camera management are:

Design prescriptions

- Allow the player to choose between computer and player-controlled systems to manage viewpoint types and camera position.
- Camera changes should not introduce disruptions in the viewing of the scenario (i.e. discontinuous jumps from one position to another), since these may confuse the player.

3.2.2 Spontaneous changes

In sophisticated designs, the status of the scenario may be modified by spontaneous changes, altering entities' dynamics and possibilities of interaction (as in `Need for Speed III`, where atmospheric phenomena may alter the status of the road, and hence the responsiveness of the cars). The only recommendation proposed by players regarding this topic is:

Design recommendation

- Use spontaneous changes as a means of introducing unexpected challenges in the gameplay, whenever they can affect interactions involving the player's token and the scenario.

3.2.3 Transitions between scenarios

Action game's virtual worlds usually have several scenarios, which makes transitions between scenarios a relevant design issue. The most common type of transition is the "complete and move to next" (represented in Figure 29, and used in games such as `Pac-Man`, `Earthworm Jim` and `Quake`), which implies unidirectional one-to-one progress through a predetermined series of scenarios, with a main transition goal G_i (which is not necessarily the goal of the game) associated with each scenario S_i, and transitions allowed only when such a goal is achieved.

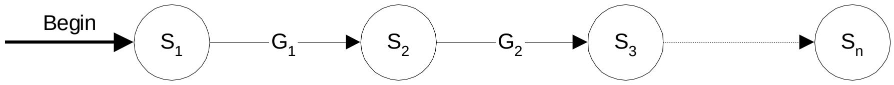

Figure 29: "Complete and move to next" transition

Other games allow less rigid transitions. As shown in Figure 30, in order to move directly from scenario S_i to scenario S_{i+k} the player has to accomplish the transition goal $G_{(i - i+k)}$, usually being allowed to move back from S_{i+k} to S_i (as in `Hexen II` and `Turok 2`, where for each scenario the transition goal is to find a teleporter which allows to move back and forth between pairs of scenarios).

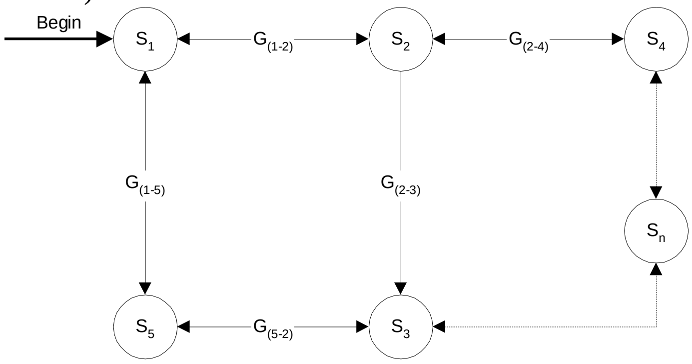

Figure 30: Semi-rigid transition

Players generally consider semi-rigid transitions as the best choice, especially if the designers follow the following recommendation:

Design recommendation

- When semi-rigid transitions between scenarios are possible, provide clues that allow the player to understand if all the tasks relevant in a specific scenario have already been performed.

3.2.4 Interactions with entities

All the formerly analyzed design aspects of scenarios are relevant to allow entities to interact with the scenario of the gaming world. Details about such interactions can be found in section 3.1.4.4.1.

3.3 Goals

Goals are the third and final global issue that affects the playability of an action game as perceived by players. Every game has a main goal. Usually, there are several interrelated goals, and their relationships lead to a hierarchy, whose main design categories are illustrated in Figure 31.

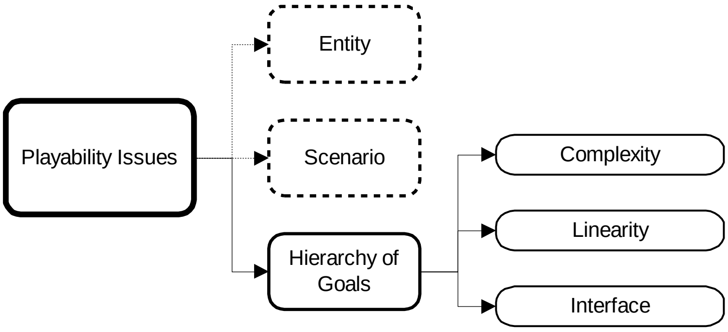

Figure 31: Design categories for the goals' hierarchy

3.3.1 Complexity

The complexity of goals may vary from game to game, and within a single game (if there are multiple goals). In simple games the main goal is to survive and complete a series of basic tasks in order to move from one scenario to the next (such as in `Pac-Man`, where the protagonist must avoid ghosts and eat pills in order to progress through an endless series of mazes). In more complex games, the main goal requires advancing through different stages of the game and achieving sub-goals. These can be the same for each stage (as in Quake, where the protagonist must kill his enemies and find an exit to the next level, until he reaches the final stage) or may vary from stage to stage (as in Turok 2, where the protagonist must solve sub-quests that are different for each stage, in order to finally defeat evil aliens). Players' preferences regarding complexity are conceptualized in the following guideline:

Design prescription

- Goals must always be understandable and unambiguous, and should not be too repetitive to avoid monotony and sustain motivation.

3.3.2 Linearity

When multiple goals must be achieved, they are usually organized in hierarchic structures that may be more or less linear. Action games are structured according to stages, each one of which has a completion goal (i.e., a final goal that allows one to consider the stage as completed and move to another one). Stages are interrelated according to dependency relationships that may be linear or non-linear.

A hierarchy of relationships (i.e., the hierarchy of goals) is said to be linear if it only allows unidirectional, one-to-one transitions between stages (as in Quake). A hierarchy is said to be non-linear if it allows some one-to-many transitions from some of its stages and/or bi-directional transitions (as in Turok 2). Figure 32 illustrates a non-linear hierarchy, with both linear and non-linear sub-hierarchies.

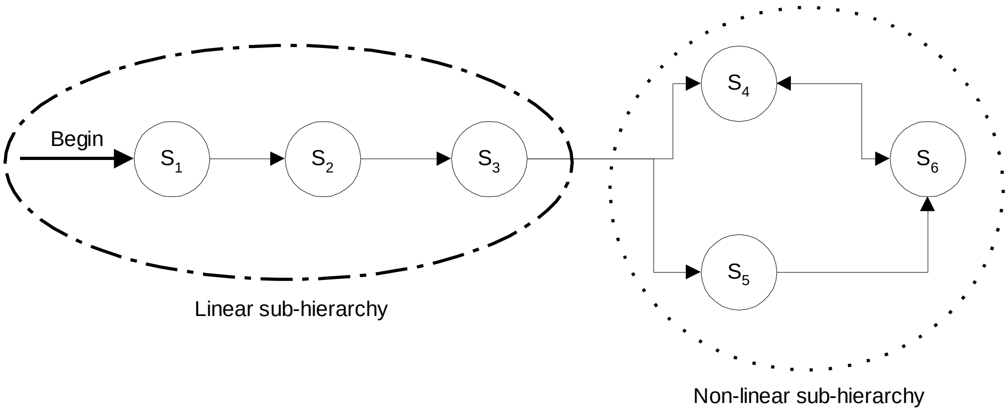

Figure 32: Non-linear hierarchy (Si: stage i)

Linearity in action games can be chronological or cause-effect (Figure 33), based on the dependencies existing between different stages.

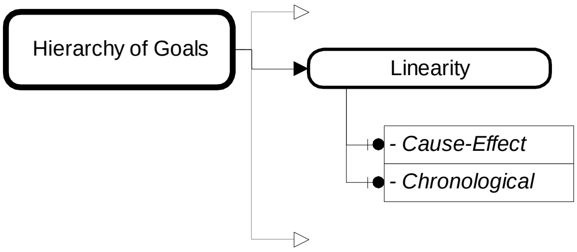

Figure 33: Linearity in the hierarchy of goals

Two stages S_1 and S_2 are defined as chronologically dependent when the player's token can move from S_1 to S_2 only upon completion of S_1, even though what it does in S_1 has no relevance in S_2 (except perhaps for finding the gateway to S_2, as in Quake). Thus, non-linearity in chronological dependencies means that the player doesn't have to complete stages according to a predetermined chronological sequence.

Two stages S_1 and S_2 are said to have a causal dependency when the completion of S_1 provides some of the means indispensable to complete S_2 (and not only to access it). Thus, non-linearity in causal dependencies means alternative ways to achieve goals and complete stages (as in Tomb Raider II).

Usually, players do not recommend strong linearity, since it is not logical to have to do things in a single, pre-determined order, without any alternatives or the possibility of moving back and forth between different stages of the game. However, non-linearity must be carefully handled, according to the guidelines proposed below:

Design prescriptions

- The relationships and dependencies between stages of alternative branches of the game, and between different stages of the same branch must always be clearly understandable by the player.
- Whenever non-linear developments are possible, it must be possible to backtrack after a decision has been made, especially if the decision is wrong and leads to negative consequences.
- Finding alternative branches in a non-linear hierarchy of goals should not be excessively time-consuming.
- Linearity is a better option in contexts in which non-linearity could make players go through visited places over and over again, trying to figure out must be done.

3.3.3 Interface

In order to achieve goals, players have to manage information about their nature and the progress made. Sometimes they could also receive helpful clues. All this information is transmitted through an interface with a contextual and an explicit component, as illustrated in Figure 34.

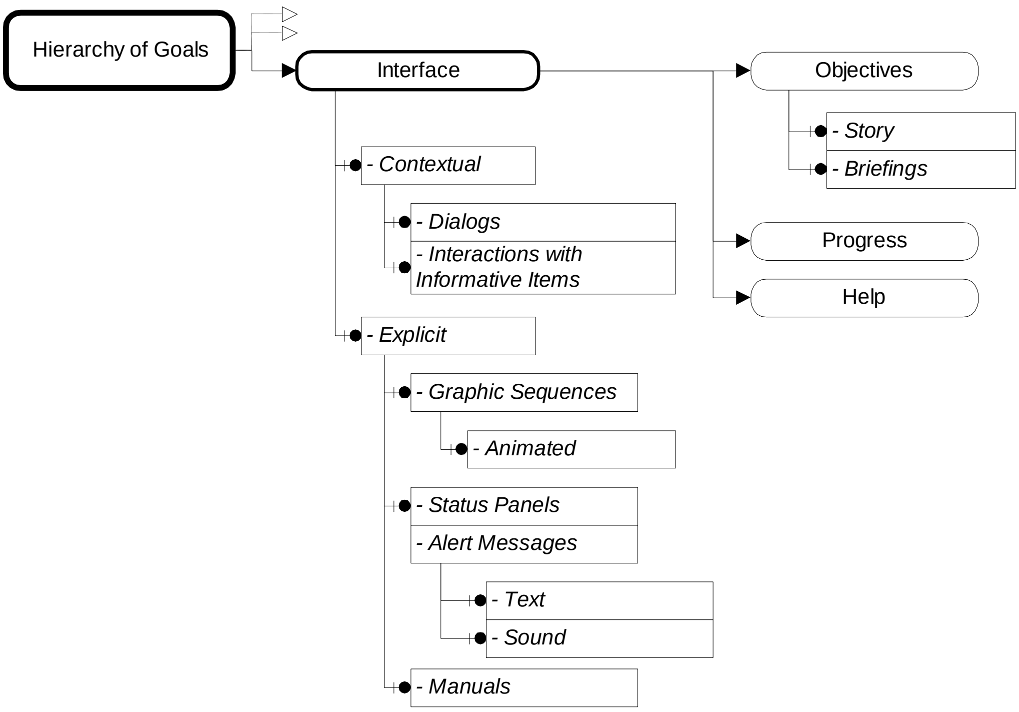

Figure 34: Interface of the Hierarchy of goals

Information regarding the goals of the game is normally transmitted through briefings and the storyline of the game. However, games don’t always provide any such information. This is not recommended, since it could make the player wonder what the purposes of the protagonist are, and make wrong decisions. Moreover, players always care about the content and presentation of such information, which leads to the importance of the following guidelines:

Design prescriptions

- The game should always provide some information regarding goals. This should be unambiguous, precise and concise.
- Briefings must clearly report the objectives of the game. It is good to use redundant means to permanently store information transmitted only once via briefings, making it available at any time during the game-playing.

- The use of a story is indispensable only when it can transmit data unavailable by other means (mainly regarding the context of the game).

Design recommendation

- The player will feel more immersed in the gaming world if the story is transmitted not only in the beginning and the end, but also during the game. By using the storytelling elements that are part of the gaming world (for instance entities that can interact with the token during the game) disruptions in the game-flow are minimized.

As for information regarding progress, whenever there are complex goals that require different steps to be achieved, it is important to provide detailed information about the status of progress, in order to allow the player to perceive whether and how he/she is progressing (as is the case in `Turok 2`). Apropos this issue, players' opinions led to the following guideline:

Design prescription

- Minimum information regarding progress should include data about failures, in order to allow the player to learn from his/her own errors.

As for help, quite often the player is aided through instructional support, the transmission of information about his/her position in the gaming world (generally through maps) or assistance provided during decision-making processes (such as customization of the virtual gaming world). Guidelines regarding help are:

Design prescriptions

- Introductory cinematics must be coherent with what the player is going to see in the game.
- Use interactive training/practice stages whenever possible, embedding them in the gameplay in order to benefit the interactivity and dynamism of the game.
- Make maps available whenever the size, visibility and rendering of the scenario don't provide enough clues to identify where the protagonist is and where known places are located.
- Maps should be easily understandable, with no ambiguities in the representation of complex elements (e.g. multistory buildings), always allowing the player to relate the information in the map with what he/she sees in his/her surroundings.

Design recommendations

- Temporarily freezing the game-flow while the player is reading a map may benefit beginners, but compromises the realism of the game.
- Providing help during customization processes regarding the gaming world may reduce game-flow disruptions.

Contextual means

The contextual interface is commonly used to provide help (through interactive training phases and the use of informative objects such as a map) or transmit information about the goals by means of briefings (as in `Grand Theft Auto`, where phone calls inform the player about what cars the protagonist of the game has to steal) and through the storyline (as in `Unreal`, where the player learns what has happened, and hence what he/she is supposed to do by listening to voice diaries found on the dead victims of hostile aliens). The most frequently used contextual means are alert messages, dialogs, informative objects and animated sequences. Guidelines regarding contextual means are:

Design prescriptions

- Voice-based messages are very immediate informative means, provided that they can be distinguished from the environmental sound, and that incoming messages are easily detected.
- Text-based messages may imply dangerous disruptions in the game-playing. Avoid them by providing options to pause the action while the player is reading the messages. The same applies to other informative objects (such as maps).
- Limit the length of non-interactive animated sequences to avoid disruptions in the game-flow.
- In order to ensure the understandability of dialogs provide at least an option for displaying subtitles. These also compensate for possible flaws in sound quality. Subtitles should also be available in different languages.
- The possibility of controlling the speech volume is indispensable to ensure that dialogs are not overwhelmed by incidental sounds or environmental music.

Explicit means

In action games, cinematic sequences, status panels, alert messages and manuals are the explicit means most frequently used to transmit information regarding the goals of the game.

Cinematic sequences and manuals are commonly employed to transmit information about the storyline of the game (as in `Turok 2` and `Crusader: No Remorse`, respectively). However, there is a common tendency to avoid using manuals, since players do not consider them a relevant part of the gameplay.

When the achievement of the goal implies different steps, information about the progress of the game is mainly transmitted through status panels (such as the mission panels in `Turok 2`).

Text and sound-based alert messages are used to provide help by giving the player hints to understand what must be done in a given situation (such as the messages that tell the player what key is needed to open a specific door, in `Doom`). Design guidelines inherent to explicit means are:

Design prescriptions

- Cinematic sequences must always be kept short (except perhaps the introductory movie) and few in number, in order to minimize disruptions in the game-flow.
- Allow the player to use alternative types and distributions of status panels to better suit his/her needs.
- Alert messages should be clear and evident (audible in the case of sound-based messages) enough to ensure that the player perceives that important information is being transmitted, and understands it.
- Complement voice-based alert messages with subtitles.

4 CONCLUSIONS

4.1 About the Relevance of This Work

The most important contribution of this work is the structure of design issues and related specific guidelines described in section 3, which, due to its inductive generality (guaranteed by the use of the *Grounded Theory* method during the data gathering/analysis process) can be considered a model (i.e. a simplified representation) that mirrors players' preferences regarding playability in action videogames.

The model complements the existing sources regarding videogame design, describing the elements that affect playability according to players' opinions and providing operational guidelines independent of specific contents (i.e. the fantasy of the game) and directly derived from players' preferences. Thus, even though the model is not a recipe for the perfect game, it can serve game designers as a prescriptive reference, enhancing their possibilities of satisfying players' preferences. This could be particularly interesting for those with little experience of videogame design, and hence with players (as may be the case of instructional designers who work on edutainment products).

Finally, the research methodology described in section 2 is an example of how a qualitative approach such as the Grounded Theory can be applied to solve a software specification problem directly focusing on end-users. The issue is particularly important since a user-centered approach can be fundamental whenever usability is a crucial element to determine the quality of a product. In fact, given the nature of most human-computer interactions problems, the application of a qualitative approach is often recommended to ensure the possibility of obtaining scientifically rigorous solutions based on empirical data that faithfully represent users' requirements (Preece et al., 1994).

4.2 Boundaries of This Research and Perspectives for Future Studies

The boundaries of this work are determined by the characteristics of the samples of games and players used during the experiments. In the first place, this research studied only the action genre. Additionally, players' age, gender and nationality clearly narrow the universe represented by the sample of participants. As a further development of this research, we consider it important to study the generality of the results proposed in this work by testing their validity against a wider universe of players and considering other game genres.

As for nationality, it would be interesting to test whether players' nationality affects their preferences. In particular, the normal trend is to distribute products worldwide. Often, specific products have different impacts on players of different nationalities (Molineux, 1998). Why is that? What are the cultural elements that could affect videogame design? Do they affect merely contextual issues (such as the ambience of specific games), or also functional aspects of the design?

As for the gender, there are reasons to believe that men and women do not share the same tastes when it comes to videogames (Klett, Rignall & Gard, 1998). Furthermore, there is clear evidence that, among children, boys and girls prefer different aspects of the gameplay, and different game dynamics (Malone and Lepper, 1987). Why is that so? Is it only a problem of specific game fantasies? Are criteria currently employed to design playability aspects of a videogame as valid for women as they are for men? Additionally, the difficulties found when looking for female participants for this research gave rise to another important question: do cultural differences between countries determine differences in the sizes of the populations of female players? If so, how?

As for the age, it is very important to study if and how the results proposed in this work are related to the age of the players. Apropos this topic, it is relevant to report that in 1998 we conducted a preliminary study to test and refine the methodology this work is based on. The experience was very similar to the one described in sections 2.2 and 2.3, and involved ten action videogames games and a sample of nineteen children aged between five and thirteen. Even though the analytical process was not as rigorous as the one described in this work, the results regarding playability were very similar to the ones described in this paper and obtained working with adults. In fact, the children belonging to the sample focused their attention on the same categories and design issues considered important by adults, although they provided less detailed information on how they relate the topics according to their preferences. Such evidence further justifies the need to study how age can affect players' preferences in terms of playability and, eventually, ambience issues.

Finally, it might be interesting to study what players learn during a gaming session. In this sense, it is important to remark that during the game-playing the player is exposed to very heterogeneous stimuli, and thus the learning processes that occur during the game-playing contribute to a very broad development of the individual, affecting a broad set of skills/abilities, and widening his/her knowledge base allowing him/her to incorporate new information regarding known and unknown issues. In such a context it might be important to analyze what learning processes are directly related to the gaming experience (for instance the learning of information related to the storyline and only useful for playing the game), and what affect more general abilities/skills (thus affecting more than the mere game-playing).

REFERENCES

CD-ROMs: The Giants Rule. *Business Week*. October 1997.

Crawford, C. (1982). The Art of Computer Game Design. [WWW document]. Available Internet http://www.erasmatazz.com.

Crawford, C. (1988). The primacy of interactivity. (WWW document). Available Internet http://www.erasmatazz.com.

Costikyan, G. (1994). I Have No Words & I Must Design. *Interactive Fantasy*, N. 2.

Glaser, B. G., & Strauss, A. L. (1967). *The Discovery of Grounded Theory*. Chicago: Aldine Publishing Company.

Hanscome, B. (1995). Interactive storytelling: The Play's the Game. *Game Developer*, December 1995.

Howland, G. (1998). The focus on gameplay. [WWW document]. Available Internet http://www.lupinegames.com.

Jones, G. (personal communication, June 1999). E-mail interview with Computer Gaming World magazine's Editor-in-Chief.

Klett, S., Rignall, J., & Gard, T. (1998). The "Babe" factor: marketing the digital goddess. *Electronic Entertainment Expo conference*. Atlanta, May 1998.

Kotler, P. (1993). *Marketing management, analysis, planning, implementation, & control*. Prentice Hall.

Lewinski, J. S. (1999). *Developer's Guide to Computer Game Design*. Wordware Publishing, Inc.

Malone, T. W. (1981a). What Makes Computer Games Fun? *Byte*, N. 6, December 1981.

Malone, T. W. (1981b). Toward a theory of intrinsically motivated instruction. *Cognitive Science*, No. 4.

Malone, T. W., & Lepper, M. R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In R. E. Snow & M. J. Farr (Eds.). *Aptitude, learning, and instruction, III: Conative and affective process analysis*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Molineux, P. (1998). Putting playability first. *Electronic Entertainment Expo conference*. Atlanta, May 1998.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human computer interaction*. Addison-Wesley.

Rinaldi, B. (1998). IDSA Reveals Survey Numbers. In *E3 Show Daily*, May 1998.

Rouse, R. (2001). *Game Design Theory and Practice*. Wordware Publishing, Inc.

Saltzman, M. A. (Ed.) (2000). *Game Design: Secret of the Sages*. Brady Publishing.

Spector, W. (1998). It's ROLE-playing, stupid! *Game Developer*, September 1998.

Strauss, A. L., & Corbin, J. (1990). *Basics of qualitative research*. Newbury Park, Ca: Sage.

Wilson, J. (1998). Play MYST-y for me. *Computer Gaming World*, August 1998.

NOTES

This is an Accepted Manuscript of an article published by Taylor & Francis Group in *Human-Computer Interaction* on 12/11/2002, available online at: http://www.tandfonline.com/doi/abs/10.1207/S15327051HCI1704_1#.U5nTsfmwJ80

Background. This article is based on Carlo Fabricatore's Ph.D. Thesis and his subsequent post-doctoral work and research as a professional game developer.

Support. This research was partially funded by FONDEF grant 1961016 (Chile).

Authors' Email Addresses.
Carlo Fabricatore: c.fabricatore@hud.ac.uk
Miguel Nussbaum: mn@ing.puc.cl
Ricardo Rosas: rrosas@puc.cl.