Entertainment Computing 2 (2011) 205–213

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

Measuring the level of difficulty in single player video games

Maria-Virginia Aponte, Guillaume Levieux *, Stephane Natkin

Centre d'Etudes et de Recherches en Informatique du CNAM, Conservatoire National des Arts et Métiers, Paris, France

ARTICLE INFO

Article history:
Received 22 June 2010
Revised 8 April 2011
Accepted 9 April 2011
Available online 22 April 2011

Keywords:
Video games
Challenge
Difficulty
Learning
Evaluation

ABSTRACT

In this paper, we discuss the interest and the need to evaluate the difficulty of single player video games. We first show the importance of difficulty, drawing from semiotics to explain the link between tension-resolution cycles and challenge with the player's enjoyment. Then, we report related work on automatic gameplay analysis. We show through a simple experimentation that automatic video game analysis is both practicable and can lead to interesting results. We argue that automatic analysis tools are limited if they do not consider difficulty from the player point of view. The last two sections provide a player and Game Design oriented definition of the challenge and difficulty notions in games. As a consequence we derive the property that must fulfil a measurable definition of difficulty.

1. Introduction

One of the fundamental issues to tackle in the design of video games is mostly referred to as *creating a well-shaped difficulty curve*. This means that one of the core element of a good game design is to make the game just as difficult as it has to be, so that the player feels challenged enough, but not too much. However, game creators cannot rely on strong tools to help them in this task, and there is not even a clear and accepted definition of difficulty as a measurable parameter. For now, game difficulty adjustment is a subjective and iterative process. Level and game designers create a sequence of challenges and set their parameters to match their chosen difficulty curve. Finding the right sequence and tuning every challenge rely mainly on playtesting performed by the designers. Playtesting is a heavy time consuming task, and it's very hard for a designer to evaluate the difficulty of a challenge he created and played for many hours. Our goal is to provide a clear, general and measurable definition of the difficulty in games. We must rely on accepted definitions of video games and works relating the games difficulty to the games quality, as perceived by the player. We present related work on automatic gameplay analysis, and then report a first experiment with a basic synthetic player. We then define difficulty, taking into account the player experience and a function of time. To conclude, we propose a way to explore the link between the player abilities and the probability to lose a challenge, providing an interesting measure for the game designer to explore the difficulty of his game's challenges.

2. Scaling the difficulty

Difficulty scaling is a fundamental part of game design [1,2]. However, this is not an obvious consequence of accepted definitions of video game. Jesper Juul has listed many of them and has proposed a synthesis [3]. We start from Juul's definition to explain why difficulty scaling is so important in game design:

> *'A game is a rule-based formal system with a variable and quantifiable outcome, where different outcomes are assigned different values, the player exerts effort in order to influence the outcome, the player feels attached to the outcome, and the consequences of the activity are optional and negotiable.'*

This definition gives a clear, precise idea of how a game system behaves, and manages to take into account the most interesting parts of the previous definitions. But for our purpose, we must explain more precisely why difficulty is a primary component of any gameplay. The fact that the player *exerts effort in order to influence the outcome*, and *feels attached to the outcome* is the core point. To point out the important components of a gameplay, and foremost the link between caring about difficulty and making a good game, it is necessary to coin a definition that leaves aside the game's dynamics structure and focuses on video games from the player's point of view.

Robin Hunicke describes a game using a *Mechanics, Dynamics and Aesthetics (MDA)* framework [4]. Mechanics are the tools we use to build a game (e.g. physics engines, pathfinding algorithm...), Dynamics describes the way the Mechanic's components behave in response to the player, and Aesthetics is the desirable emotional responses evoked to the player. Of course, the design goals is the Aesthetics, that is to say the player's emotions. We argue that

* Corresponding author.
E-mail addresses: maria-virginia.aponte_garcia@cnam.fr (M.-V. Aponte), guillaume.levieux@cnam.fr (G. Levieux), stephane.natkin@cnam.fr (S. Natkin).

1875-9521/$ - see front matter © 2011 International Federation for Information Processing Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.entcom.2011.04.001

the difficulty of challenges greatly influences video game's aesthetics and thus plays a central role in game design.

Umberto Eco's book *The open work* is a fundamental research about interactive art's aesthetics [5]. Umberto Eco states that when we face a piece of art, we are interpreting it, seeking patterns and looking for information. Depending on our culture and knowledge, we will find something to grab on within the stimulating field of the piece of art. But then we will go further, and find another interpretation and feel lost for short moment, while shaping our new pattern. Moreover, when a piece of art is interactive, the aesthetic value comes both from the tension resolution and from the fact that this resolution is a consequence of our choice. Assuming that a video game is an open work we can propose a similar analysis. Every time the player faces an obstacle, he gets lost for a few seconds. Then he finds and chooses a pattern, presses the right buttons, and takes pleasure both from resolving a tension and from making a choice. Thus, we can draw from Umberto Eco's work that in video games, challenge is fundamental because it creates tension situations that the player has to solve and the opportunity of meaningful choices.

Related work on video game player's enjoyment support our analysis and place challenge at the center of video game's aesthetics. In his book *A Theory of Fun for Game Design*, Ralph Koster states that we have fun playing games when we discover a new pattern, i.e. a strategy that we apply to overcome a challenge [6]. Sweetser and al see challenge as one of the most important part of their *Game Flow* framework [7]. Yannakakis et al. measure player enjoyment from challenge, besides behavior and spatial diversity [8].

Mihaly Csikszentmihalyi's Theory of Flow, that researchers have applied to video game as a measure of the player's enjoyment, helps us to make a link between the difficulty of a challenge and the player's enjoyment [9,10,7]. A player is in a *Flow* status, and thus enjoying the game, when the task is neither too hard nor too easy. It is thus not enough to create tension situations and to give the player choices to resolve this tension. A good game design must accurately scale the difficulty of a challenge to have a tension level that leads to the player's enjoyment. Thus, a definition of a game from the Aesthetic point of view and centered on challenges could be:

> *'Regarding challenges, the Aesthetics of a game is created by tension-resolution cycles, where the tension is kept under a certain threshold, and where the resolution of a cycle depends on the player's choices.'*

This definition does not take into account every aspect of game aesthetic but is focused on challenge, that most studies consider as a core component of game's aesthetics. Tension situations that the player seeks and try to solve have been created by the game designer and the amount of tension they deliver directly stems from their complexity. As a result, difficulty scaling is a central task of a good game design. Games already propose multiple difficulty levels [1], and sometimes even Dynamic Difficulty Adjustment [2], manipulating some specific parameters of the gameplay in real time [4], or automatically scaling the game AI capacity [11]. But whichever difficulty scaling method the game designer uses, he must still tune them properly. It is sometimes really hard to guess to which extent a change in a low level parameter will just make the game a bit harder or dramatically change the gameplay [1], and tuning is one of the most time consuming area in game AI development [12]. This is the design process that we want to shorten by providing tools that will help game designers evaluating the impact of any difficulty scaling parameter on the final difficulty curve. It's then fundamental to provide game designers with strong tools and a definition of difficulty as a measurable parameter.

3. Related work: testing with a synthetic player

Our goal is to evaluate a parameter or a set of parameters that can be considered as a measure of a game difficulty. There are two theoretical approaches to evaluate such a parameter. The first way is to find, according to the game structure, a mathematical expression of the parameter and to solve the corresponding equations. The complexity of a game and the notion of difficulty tend to show that this approach is not practicable. The second solution is to experiment the game and measure the parameter. To experiment the game, we may use either a real or a synthetic player. The main advantage of using a real player is that it demonstrates human behaviors. In counterpart he plays slowly, becomes tired and his behavior is only known through the game interface. The synthetic player is tireless, plays quickly and his behavior can be fully understood. The design of the synthetic player allows to simulate some foreseen behavior of a real player (risky or careful, for example) and some simple learning techniques.

Gameplay testing has already been the subject of many interesting researches. Alasdair Macleod studied gameplay testing of Perudo, a bidding dice game, simulating plays with a multi-agent system [13]. He wanted to modify Perudo's gameplay to make it more fair, and added a rule which he thought would help losing players to stay in the game. By running the experiment and analyzing the modified game, he obtained the counter-intuitive result that the rule was not helping the players at all. These results shows that self-play testing can help testing gameplay modifications.

Neil Kirby analyzed Minesweeper, replacing the player by a rule based AI [14]. Each rule was related to a different play complexity. He found out that Minesweeper was surprisingly not as hard as he supposed it to be, as the most part of the board was often solved using only the very simple rule. These results point out that automated techniques can provide interesting approaches to study video game difficulty.

Both Perudo and Minesweeper are simple games, but automated analysis can also be applied to complex off-the-shelf games. Bullen et al. used Unreal Engine (Epic Games) and created a gameplay mutator providing sensors to log useful game events [15]. They tested Unreal Tournament 2004 (Epic Games) using partial and fully automated testing (i.e. both during player vs AI and only AI games). They pointed out that fully automated tests had to be done with a specific AI, because standard AI was not aggressive enough. The fact is that standard Unreal Tournament AI has been created to entertain the player, not to mimic his behavior, and thus is not able to fully explore the gameplay. Recently, Lankveld et al. proposed to analyze a game difficulty using incongruity, the distance between the actual dynamics of the game and the mental model the player has built [16]. They plan to infer the complexity of the player's mental model, and thus the difficulty of the game, by monitoring his actions. These works show that, to be useful, a synthetic player must simulate in some way a real player.

Automated game analysis can be done at several levels. Nantes et al. distinguish Entertainment Inspection (i.e. gameplay testing), Environment Integrity Inspection (i.e. Sounds, graphics related issues) and Software Inspection [17]. Their system targets Environment Integrity Inspection, using Computer Vision, and especially corner detection to detect aliasing issues in shadows rendering. This is a complementary approach to the one we propose, and Nantes et al. acknowledge the need of analysis tools at every inspection level.