

Evaluating dynamic difficulty adaptivity in shoot'em up games

Bruno Baère Pederassi Lomba de Araujo and Bruno Feijó

VisionLab/ICAD

Dept. of Informatics, PUC-Rio

Rio de Janeiro, RJ, Brazil

baere@icad.puc-rio.br, bfeijo@inf.puc-rio.br

Abstract—In shoot'em up games, the player engages in a solitary assault against a large number of enemies, which calls for a very fast adaptation of the player to a continuous evolution of the enemies attack patterns. This genre of video game is quite appropriate for studying and evaluating dynamic difficulty adaptivity in games that can adapt themselves to the player's skill level while keeping him/her constantly motivated. In this paper, we evaluate the use of dynamic adaptivity for casual and hardcore players, from the perspectives of the flow theory and the model of core elements of the game experience. Also we present an adaptive model for shoot'em up games based on player modeling and online learning, which is both simple and effective.

Keywords—Artificial Intelligence, Player Modeling, Games, Adaptivity, Dynamic Difficulty Adaptivity, Dynamic Difficulty Adjustment

I. INTRODUCTION

Being part of human culture [26, p. 6], games are sought by players for the enjoyment found in overcoming its challenges [32, pp. 38–42]. The flow theory by Csikszentmihalyi [16] shows that there must be a balance between difficulty and player skill for this enjoyment to happen [32, pp. 98, 128].

One kind of game known for being a niche game is the shoot'em up¹ variant. Such games usually present difficult challenges demanding high skills from the player, which can keep some kinds of players off the experience. This genre of video game is quite appropriate for studying and evaluating dynamic difficulty adaptivity in games that can adapt themselves to the player's skill level while keeping him/her constantly motivated.

In this paper we present an efficient and simple implementation of a dynamic difficulty adaptivity system for balancing game difficulty and evaluate it under the flow theory [16] and the core elements of the game experience [5].

II. PREVIOUS WORK

Adaptivity in games is not a new theme, although it's certainly trendy [34]. Adaptivity can be identified in many traditional games, such as Go and golf [39], but one of the first documented uses of difficulty adaptivity in video games can only be found in the late 80's, such as in the shoot'em

up game called Zanic [12], [13]². More recent games that implement some kind of difficulty adaptivity are Mario Kart 64 [40], Max Payne [20], the Left 4 Dead series [53], [54], and the GundeaddliGne series [2].

Charles et al. [9], [10] proposed a framework for creating dynamic adaptive systems for video games including the use of player modeling for the assessment of the system's response. Hunicke et al. [27], [28] proposed and tested an adaptive system for FPS games where the deployment of resources, and items is based on player performance. Ibáñez and Delgado-Mata [29] tested an adaptive Pong for two players with positive results, for both the more and the less skilled players. In Infinite Adaptive Mario [56], a variant of Super Mario games was created where the next stage of difficulty is determined by the player performance in the previous stage.

Other works using machine learning include: Spronck's dynamic scripting [50], an adaptive technique based on mixing AI scripts that define the non-player character's strategies; Demasi and Cruz's work in using fuzzy rules, fuzzy state machines, and genetic algorithms to adapt the enemy AI [17]–[19]; Noon's neuroevolutionary controls based on player modeling [41]; The use of a M5P classifier and top culling by Machado et al. [35]; Yannakakis et al.'s work on player modelling support for adapting the game [59]–[61]. Also there are proposals that are inspired on models of e-learning and e-commerce [47].

In this paper, we are interested in evaluating the use of dynamic difficulty adaptivity in games from a more systematic view of the game experience [7], [14], [33]. In order to accomplish this task, we developed a shoot'em up game and a simple although effective adaptive model based on player modeling and online learning. We used 35 subjects, fairly distributed amongst casual/hardcore players and male/female ones. Also we used the framework proposed by Charles et al. [10]. As far as we are aware, no other work in the literature has such a systematic and effective approach to analyze difficulty adaptivity in games. We consider that shoot'em up games are the most appropriate genre of game for studying and evaluating dynamic difficulty adaptivity. Moreover this kind of video game is quite appropriate to test effective algorithms. We have no intention to generalize the results to other types of video game.

¹For more information about the shoot'em up games genre, we recommend the following sites <http://www.racketboy.com/retro/shooters/shmups-101-a-beggins-guide-to-2d-shooters> and <http://www.racketboy.com/retro/shooters/games-that-defined-the-shmups-genre>.

²Compile was a Japanese video game publisher dissolved in 2003, but the last version of Zanic (Zanic x Zanic) for PS1 can be found in Sony's online service, Playstation Network, for both PS3 and PSP.

III. DEFINITIONS

For delimiting the focus of this work, some concepts such as game, player, and flow must be well defined and limited to video games.

A. Game

Game has been defined by many authors [15], [26], [45], [46]. Juul [31] defined games as a formal system of rules where the player is emotionally linked to the result of its effort in working with this set of rules (playing the game). In this paper we use this definition of game. Koster brings the concept of fun in games [32, p. 40], when players understand and dominate the challenges of the game, which are presented and identified as patterns. With fun also comes boredom, which is defined as the lack of new patterns (or challenges) or the difficult being too high or too low [32, p. 44].

There is also the concept of anti-Buddhism in games, introduced by Poole [44], where “having more lives” is a good thing in games as there is more learning possibilities; so the fact of “having to reincarnate” many times is not necessarily a bad thing. In fact, Xavier [58, pp. 216–217] cites that players will sacrifice one of those lives by their own will for the knowledge gained in such way.

The above-mentioned concepts about games help us to understand that one of the factors to maintain player motivation in playing the game is the continuous challenge to player’s skills, but not so high as to put the player off the experience and not so low that he/she will be bored. The challenge must be proportional to the player’s skills. This is well supported by the flow theory (section III-B).

The concept of difficulty in games can be derived from the challenge-skill relationship, that is: the higher the skill demanded to solve the challenge, the higher is the difficulty. However, this is a quite subjective concept, because the necessary player’s skills can be hard to quantify. This type of relationship is derived from the concept of flow. In games, it is common to represent difficulty as a scale of power, speed or number of enemies or puzzles, or in information available to complete an objective.

B. Flow

The concept of flow defined by Csikszentmihalyi [16], relates difficulty in a task to the skills of the performer and a state of mind during the execution of such a task in which the performer is so absorbed in the experience that he/she can even lose track of time. Csikszentmihalyi [16] says that flow was developed by humans as a way of recognizing patterns of action, which relates to Koster’s [32] definition of how players achieve fun in games. Cowley et al. [14] have too observed the close relationship between flow and games, correlating flow elements to game-play elements.

Even though, not every person is capable of achieving the flow state. As Schell [48] states, the activity must have clear objectives, there must be no distractions, the activity must give direct feedback to the performer, and the challenge must be continuous [48, p. 119]. The performer must have the skills necessary to the task and proportional to the challenge, and must have an autothelic personality, that is, the performer must

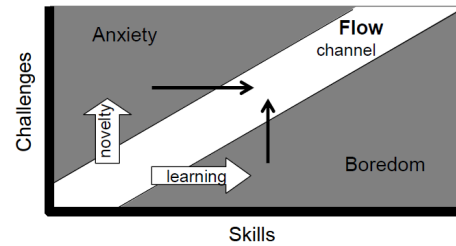


Fig. 1. Flow channel and the proportionality of skill and challenge for achieving flow, extracted from [14].

be able to seek the flow state. Figure 1, extracted from [14], shows this relationship.

C. Player

Being the player the one who interacts with the game, what does the player look for in a game? Huizinga [26] and Koster [32, pp. 40–44] put that the player seeks fun in games. The different motivations of how the player can achieve fun have generated various forms of player classification, ranging from demographic classifications [42, pp. 56–70] and [32, pp. 4–10 and pp. 48–50] to psycho-types such as Bartle’s player types [4]. An effective classification is to divide players into hardcore and casual ones [42, p. 54].

According to Fortugno [21], casual players and hardcore players differentiate from the set of skills, the tolerance to failure and repetition of tasks, and different levels of auto-motivated exploration [21, pp. 144–146]. These differences in their motivation make it hard for game designers to design a particular experience for a wide range of players. For instance, a game designer can design a game that suits hardcore players for its difficulty but that may be not approachable by casual players. Even dividing the game into layers of difficulty, such as easy, normal, and hard, this discrete approach may still not suit a great variety of players.

IV. DYNAMIC DIFFICULTY ADAPTIVITY

Players should be provided with the right amount of challenges, in such a way that the game does not generate boredom nor anxiety and the difficulties are adaptable to the player’s skills. Some proposals of adaptive systems that provide good playing experiences are presented in section II. In the old Zanic [12] game, this system was called automatic level of difficulty control. This balance between skill and challenge is defined by Novak [42, p. 202] as a state when players perceive the game as consistent, just, and fun.

As players differ from one another, adaptive systems have been used to personalize the gaming experience to each player, adjusting games directed by objectives that can be identified, measured, and influenced [34]. These systems generally use a dynamic factor such as the player’s skills, which evolves as the player progresses through the game.

According to Andrade et al. [1], dynamic adaptivity in games must satisfy three requirements: identify and adapt itself to player’s skill level the quicker as possible; perceive and register the changes in player performance; keep the

game behaviour as discrete and credible, so the player cannot perceive the adaptive system.

Adaptivity can be done in two ways [34]: online or offline. Offline adaptivity occurs when the data is acquired or used to adjust the game parameters before the game-play is active. Online adaptivity occurs in real-time during game-play.

According to Chen [11], the use of automatic dynamic difficulty adjustment alone is not capable of leading players to experience flow. The main problem is that automatic systems take control away from the game designers. Chen's [11] proposal to this problem is to let the player decide when and how to change difficulty based on game-play choices planned by the designer. Thatgamecompany's *fIOW* [52] is Chen's game reference implementation for his proposed active dynamic difficult adaptivity.

A. Adaptivity vs. Adaptability

There is a difference between adaptivity and adaptability. Adaptability is related to the ability of the system to be adjusted and modified by a user, while adaptivity is the ability of the system to modify itself to suit the user [23]. Almost all games offer some kind of adaptability, be it the adjustment of difficulty, number of lives or other options.

It is expected from the game to offer a crescent level of difficulty as the player progresses through the game, but the challenges and difficulty planned by the game designer for that experience and the parameters set in the beginning of the adaptable game may not suit the player [23]. An adaptive game would circumvent this flaw by allowing the player to experience challenges with difficulty proportional to his/her performance.

B. Player modelling

Player modelling is a technique based on educational adaptive systems [25, p. 557]. It is used to infer higher-order attributes from the player using data gathered from game-play or before the game-play, so that the player can be classified using a certain algorithm suited to the domain of the application [14]. It can be also known as opponent or adversarial modelling [3], [55].

This information gathered can be very diverse: directly related to the player, such as its preferences, previous knowledge, age, sex, and in-game tactics. or related to its game character, such as playtime, lives, accuracy, weapon choices, etc. These data can then be used to adapt the game AI to the player [25] after translated to a specific modelling.

There are many works in player modelling, such as fuzzy models by Demasi and Cruz [17], Charles et al.'s player modelling framework [9], [10], Missura and Gärtner's supervised learning [38] and the use of neural networks [43], [59]–[61]. Also there are proposals for player modelling classification, such as the taxonomic works of Machado et al. [36], [37], and Smith et al. [49].

The complexity of the player model and the information that defines it depends on the quantity of detailed information necessary to transform the model into useful game data, so the modelling is dependent and almost exclusive to the game

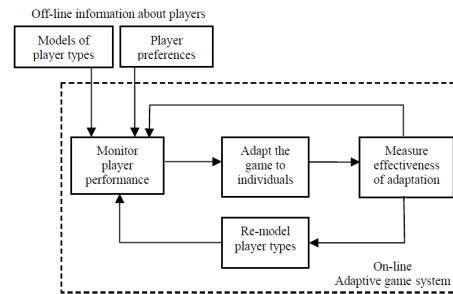


Fig. 2. Framework proposed by Charles and Black, extracted from [9].

in which it is used, as Houlette says: choosing the right parameters for player modelling ends up being more art than science [25, p. 565].

C. Charles and Black's framework

Charles and Black proposed [9] a framework for developing adaptive systems based on player modelling. Their main contribution is formalizing the necessity of a player modelling related to the game's adaptive performance, allowing the effectiveness of intelligent agents to be measured by the evolution of the player model, correlated to the player's frustration level.

If there is no perceived advancement of player's performance or reduction of player's frustration, the player may have been incorrectly classified or the initial player modelling is not valid and must be changed.

The framework is composed of four main aspects [9]: player modelling, adaptive game environment in response to player's necessities, monitoring of the effectiveness or compatibility of any adaptation, and remodelling or dynamic classification of the player.

The entry data for the framework is a set of player models and the player preferences (see figure 2), so a initial model about the player can be built and identified as the off-line part of the adaptive system. The system feedbacks itself during game-play through the evaluation of the changes made to the model and the player performance, comprising the online part of the adaptive system.

D. Adaptivity challenges and problems

Lopes and Bidarra [34] show some of the problems faced by adaptivity. One of the problems is that to adapt the game to the player's motivation it is necessary to determine player's expectation, quantify it so it can be measured, compared and adjusted, process the signals acquired from player input and performance to make the correct adjustments. Part of these problems can be addressed with player modelling.

Mario Kart 64 [40] is a classic example of rubber-banding [9], [10], [27]. Its adaptivity reduces other racers speed when the player is performing badly and increases it when the player is performing well. A player that detects this adaptivity can abuse it by purposely performing badly and in the last laps performing well, so the system takes time to adapt and increase speed, letting the player advance to better positions.



Fig. 3. A typical situation the player will face

In Max Payne [20] we find an example of anti-Buddhism [44] break. If the player dies too much at a section, enemies position and quantity can be adjusted. The anti-Buddhism break occurs because each failure gives the player a chance to memorize and adapt its strategy to the challenge, but removing enemies breaks the suspension of belief and what the player knew and should be deterministically consistent. A similar problem can be noticed with Infinite Adaptive Mario [56] as when the player dies too much in a stage, it is substituted for an easier one.

In The Elder Scrolls IV: Oblivion [51] the adaptive system used scales enemies power with the player level, which generates verisimilitude problems within the game world: questions such as "why these road bandits wear better equipment than mine?" and logistics problems such as not having the appropriate equipment to deal with different enemies, as the measure of power used is the level that advances with skill use. A player can be of high level by levelling skills, not necessarily having achieved magical equipment and spells that would deal with higher level enemies.

Other challenge that Lopes and Bidarra [34] identified was to support adaptivity mechanisms in a way that they can be reused independently from the game genre and domain.

V. METHODOLOGY

We developed an implementation of Charles and Black's framework [9] for player modelling and dynamic adaptivity as a library, applying it in a shoot'em up game developed for this research. We used C++ and Lua [30] for programming and scripting and ClanLib³ as the game engine. All assets used are under GPL or Creative Commons licenses.

In the game, the player controls a starship that must destroy alien invaders and survive through several waves of enemies. Each enemy has an attack pattern that the player must learn in order to fight back. Each time the player is hit, he/she loses one of the five lives that starts with. If the player lives reach zero, game will be over. If the player can survive all the eight waves of enemies, he succeeds in defeating the invaders. Figure 3 represents a typical situation in the game.

Two versions were developed: an adaptive version, using the implementation of Charles and Black's framework [9], and a non-adaptive version. Both versions allow the player

³<http://clanlib.org>

to choose the initial difficulty, used as entry data for the adaptive algorithm. For the adaptive game, the difficulty varies accordingly to player performance. For the non-adaptive game, the initial difficulty is the same used for each wave of enemies.

A. Adaptive algorithm

The modelling of the difficulty level of the game and its adjustment takes in account that each enemy can be represented by a set V of behaviour variables:

$$V = \{speed, shotDelay, halfRange\} \quad (1)$$

These variables are set to their initial reference values established by the programmer (equations 2a, 2b and 2c). For instance, we used $speed_0 = 300$, $shotDelay_0 = 900$ and $halfRange_0 = 200$.

$$speed = speed_0 \quad (2a)$$

$$shotDelay = shotDelay_0 \quad (2b)$$

$$halfRange = halfRange_0 \quad (2c)$$

These variables are the base for higher-level behaviour of the NPCs, such as agility and accuracy. Speed is directly related to movement calculations of the enemy ship. Shot delay affects the enemy rate of fire, so that a lower level represents a high rate of fire. We define half range as the area of threat that determines the accuracy of the enemy, so the lower its value, the higher the accuracy of the enemy and consequently more experienced must be the player to face this enemy.

A way to adjust the game to a particular level of difficulty is using a multiplication factor for the behaviour variables. In our game, we considered the following types of players based on a set of difficulties:

$$Types = \{easy, medium, hard\} \quad (3)$$

and we defined a difficulty multiplier $m(type)$, which is neutral (i.e., equal to 1.0) for the medium type. In our game, we considered the following multipliers:

$$m(easy) = 0.85 \quad (4a)$$

$$m(medium) = 1.0 \quad (4b)$$

$$m(hard) = 1.2 \quad (4c)$$

Based on these concepts, the model adjustment of the enemies is done applying the type multiplier on the enemies behaviour variables, generating a new set of values, as described in algorithm V.1.

Algorithm V.1 function `adjust(type)` return values of V

$speed \leftarrow speed_0 \times m(type)$

$shotDelay \leftarrow \frac{shotDelay_0}{m(type)}$

$halfRange \leftarrow \frac{halfRange_0}{m(type)}$

$newV \leftarrow \{speed, shotDelay, halfRange\}$

return $newV$

Player remodelling is based on a definition of a C set of n characteristics of player performance we called traits (see equation 5). At the end of each wave, the trait values are computed and the new player model is evaluated against the previous model.

$$c_i \in C / c_i \in [-1, 1], i = 1, n \quad (5)$$

Each player model is delimited by minimums and maximums for each trait c : $c_{i,min}^{type}$ e $c_{i,max}^{type}$. For example, $c_{2,min}^{medium} = 0.3$ e $c_{2,max}^{medium} = 0.6$, as illustrated in table I. Therefore, the minimum and maximum values that define each player model are defined as:

$$MIN^{type} = \sum_{i=1}^n c_{i,min}^{type} \quad (6a)$$

$$MAX^{type} = \sum_{i=1}^n c_{i,max}^{type} \quad (6b)$$

TABLE I. PLAYER MODELS IMPLEMENTED

	Easy		Medium		Hard	
	Min	Max	Min	Max	Min	Max
Accuracy	0.0	0.3	0.3	0.6	0.6	1.0
Lives variation	0.6	1.0	0.3	0.6	0.0	0.3
Enemies per wave	0.0	0.3	0.3	0.6	0.6	1.0
Enemies total	0.0	0.3	0.3	0.6	0.6	1.0
Total	0.6	1.9	1.2	2.4	1.8	3.3

The adaptive algorithm we developed is based on the framework [9] and the adaptivity method proposed by Houlette [25]. It is described in algorithm V.2. The alpha factor used determines how much the system learns from the current observation [25, p. 560] and was inspired in Widrow and Hoff's delta rule for online learning of neural networks [57, pp. 123–134].

Algorithm V.2 Adaptive algorithm

```

 $\alpha \leftarrow \text{learningRate}$ 
 $type_0 \leftarrow \text{initial type informed by the player}$ 
 $c_i \leftarrow \frac{(c_{i,min}^{type_0} + c_{i,max}^{type_0})}{2}$  {i.e., the average of the standard
performance  $type_0$  for each trait  $c_i$ .}
 $V \leftarrow \text{initial state of behaviour variables}$ 
for all waves do
   $c_{i,obs}$  is the perceived trait value  $i$ 
   $c_i \leftarrow c_i + \alpha \times (c_{i,obs} + c_i)$  {i.e., updates each trait by
LMS.}
   $performance \leftarrow \sum_{i=1}^n c_i$ 
  if  $performance \in [MIN^{type}, MAX^{type}]$  then
     $newModel \leftarrow type$ 
  else if  $currentModel \neq newModel$  then
    Remodel player:
     $V \leftarrow adjust(currentModel)$ 
  else
    Maintains current model
  end if
  Store wave's statistics
end for

```

B. Adaptive system

The proposed adaptive system consists of an intelligent agent, the AIManager, that perceives the game environment through the performance data of the player and modifies the environment altering the NPCs behaviour variables, aiming for a difficulty level that suits the player's performance, i.e. a difficulty level that provides challenge to the player without being too easy (identified by a high player performance, such as high accuracy, low lives variation, or number of enemies defeated) nor too hard (identified by a low player performance, such as high number of deaths - lives variation - and low accuracy, or low number of enemies defeated in a wave).

We chose not to alter the player's characteristics (such as speed and rate of fire) because these changes are easier to spot, as the player is in constant control of its character. Slightly changing the enemies characteristics is a way that can be a little harder to perceive the changes.

Figure 4 shows our implemented system. Each enemy implements an AI Agent interface, through which the AIManager manages changes to their characteristics, at the end of each wave that a change in the player model was observed, via the updateAgents method that calls an updateStats method implemented by the client code (the game). The updateStats method adjusts the enemy's characteristics to suit the current player model. Algorithm V.3 shows how the player model is updated using a comparison method implemented by the client code, exemplified in algorithm V.4.

Algorithm V.3 function AIManager.update()

```

 $result \leftarrow 0$ 
for  $playerModelIterator \leftarrow playerModels.begin()$  to
 $playerModels.end()$  do
   $result \leftarrow$ 
   $currentObservedModel.(playerModelIterator)$ 
  if  $result < 0$  then
     $continue$ 
  else if  $result = 0$  then
     $currentReferenceModel \leftarrow playerModelIterator$ 
  else
     $result \leftarrow currentReferenceModel.compare$ 
     $(playerModelIterator)$ 
    if  $result < 0$  then
       $currentReferenceModel$ 
       $playerModelIterator$ 
    end if
  end if
end for
 $currentObservedModel.setName$ 
 $(currentReferenceModel.getName())$ 
 $updateAgents()$ 

```

As in table I, there are noticeable intersection pairs: Easy-Medium and Medium-Hard. As the player models are sorted in the comparison vector, this intersection allows a lower ascension in difficulty levels and a faster descent with the intent of reducing player's frustration with a high difficulty.

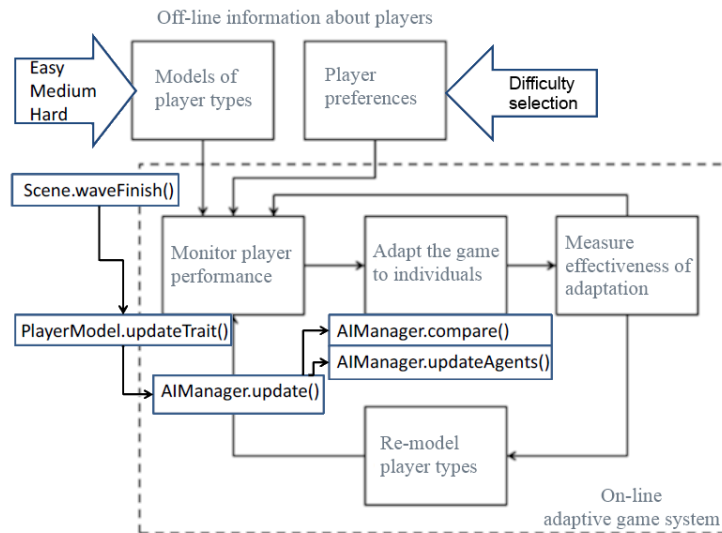


Fig. 4. Overlay of our dynamic difficulty adaptive system on Charles and Black's framework [10]

Algorithm V.4 function `PlayerModel.compare(comparable)`

```

total ← 0
totalMin ← 0
totalMax ← 0
for  $i \leftarrow 0$  to  $numOfTraits$  do
  total ← total +  $getTraitValue(i)$ 
  totalMin ← comparable.getTraitMinimum(i)
  totalMax ← comparable.getTraitMaximum(i)
end for
if total < totalMin then
  return -1
else if total > totalMax then
  return 1
else
  return 0
end if

```

C. Evaluation of the player experience

For evaluating the player experience, we selected 35 persons to test the game. These players were told they would play two versions of a same game, but they were not told of any difference between each version. This was done so the players could then tell if they felt any or no difference between each version. The playtesting followed Fullerton's recommendations [22, pp. 252–269].

We conducted the experiment in three steps. Firstly, the players answered a demographics questionnaire to assess information about player's age, previous experience with games (game genres known, weekly hours of play and if he/she considered itself hardcore or casual player). Secondly, the players played a version of the game and answered a post-

game experience questionnaire and then played another version of the game and answered the same post-game experience questionnaire relative to that version. Player's performance was logged for each game played. For each player the starting version was changed to minimize learning bias. Lastly, an interview was conducted to assess subjective and qualitative data about the game experience, according to Hoonhout [24, pp. 72–73].

Post-game experience questionnaires have been used in previous works [6], [7], [29], [35]. In our work, we decided to use the CEGE framework by Cávillo-Gómez et al. [7], [8]. We used a 7-point Likert scale that assesses the core elements of gaming experience to detect which version gave the player the best experience in terms of these elements.

The CEGE framework [7] states that the interaction between player and video games is analogous to puppetry manipulation [8]: initially there is the approximation between player and game and this interaction involves to a point that the game being played is the result of the player's actions. This relation between puppetry and video game can be hierarchically structured as: (1) the core elements of the gaming experience; (2) constructor elements that allow the perception of the core elements; (3) observable elements of the process that are consequences of the constructor elements. Figure 5 shows this hierarchy. The questionnaire was created using observable variables (that can be directly measured). Figure 6 shows the relationship between the core elements and the questionnaire variables.

To evaluate the game experience, we used two sets of factors (latent variables) from the CEGE framework divided into scales 1 and 2, and a total of 38 questions. Table II shows to which variables each question is related.

Set 1 contains Enjoyment, Frustration, CEGE, Puppetry

Latent variables	Puppetry			Videogame		Core elements
	Control	Ownership	Facilitators	Game-Play	Environment	Constructor elements
	Small Actions	Big Actions	Time	Rules	Graphics	
Observable variables	Controllers	Personal Goal	Aesthetic Value	Scenario	Sound	Observable elements
	Memory	You but not You	Previous Experiences			
	Point of View	Rewards				
	Goal					
	Something to do					

Fig. 5. Core elements of game experience and the hierarchical structure of elements relationship. Adapted from [8].

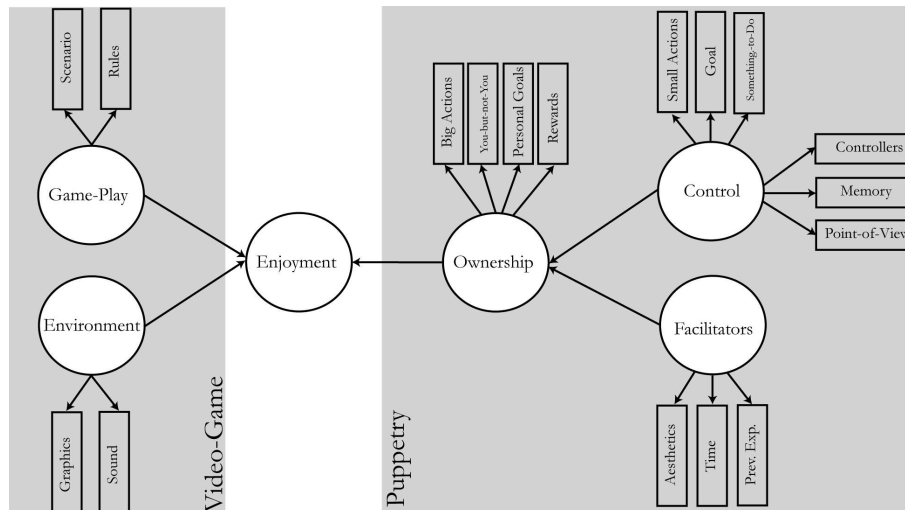


Fig. 6. Relationship between variables from CEGE questionnaire, Relationship between variables from CEGE questionnaire where circles are latent variables and rectangles are concrete and observable variables from [8, p. 64].

and Video-game defined as follows: enjoyment refers to the fun in playing; frustrations refers to the frustration in playing; CEGE refers to both puppetry and video-game; puppetry refers to the sensation of control and dominance over the game; video-game refers to playability and environment, defined by graphical and sound elements. In this set, puppetry and video-game are correlated to enjoyment, and if CEGE is present then frustration must be low and not correlated [8, p. 65], but there is no guarantee that enjoyment is positive.

Set 2 contains Control, Facilitators, Ownership, Environment and Game-play defined as follows: control refers to the sensation of control over the game, that is, the player is making the game answer to its actions; facilitators are subjective elements such as previous experiences with similar games; ownership refers to the sensation that the game is an extension of the player; environment is related to the game graphics and sound elements; game-play refers to the games rules and story. This main variables of this set that correlate to fun or enjoyment are environment and game-play.

TABLE II. RELATIONSHIP BETWEEN QUESTIONNAIRE QUESTIONS AND GAME EXPERIENCE FACTORS, ADAPTED FROM [8, P. 65].

Items	Factor
1, 4, 5	Enjoyment
2, 3	Frustration
6-38	Core Elements of Game Experience
6-25, 38	Puppetry
26-37	Videogame
6-12, 25, 28	Control
13-18	Facilitators
19-25	Ownership
26-31	Environment
32-37	Game-play

VI. RESULTS

Table III summarizes the testers by sex and their classification as hardcore or casual. One of the participants decided to classify itself as non-player. We considered this case as a casual player for the rest of the analysis.

The results of the questionnaires are summarized in table IV for the hardcore players and table V for casual players. As each question was graded in a 7-point Likert scale, we summed the contribution of each question to each set of latent variables

TABLE III. PARTICIPANTS SUMMARY - SEX, CLASSIFICATION

Participants					
Total	Male	Female	Casual	Hardcore	Non-player
35	16	19	18	16	1
%	46%	54%	51%	46%	3%

TABLE IV. COMPARISON OF CEGE SCALES FOR HARDCORE PLAYERS

Comparison of CEGE scales for hardcore players						
		Version 1		Version 2		Difference %
Factors		Sum	Mean	Sum	Mean	
Scale 1	Enjoyment	281	5,8542	285	5,9375	-1,40%
	Frustration	74	2,3125	85	2,6563	-12,94%
	CEGE	2925	5,5398	2880	5,4545	1,56%
	Puppetry	1775	5,2827	1756	5,2262	1,08%
	Video-game	1150	5,9896	1124	5,8542	2,31%
Scale 2	Control	866	6,0139	859	5,9653	0,81%
	Facilitators	478	4,9792	477	4,9688	0,21%
	Ownership	529	4,7232	511	4,5625	3,52%
	Environment	592	6,1667	566	5,8958	4,59%
	Game-play	558	5,8125	558	5,8125	0,00%

considered according to table II. Each table presents the results of the sum and mean of the questions considered. Version 1 refers to the adaptive version of the game, and version 2 refers to the non-adaptive version of the game.

In table IV it is clear that the adaptive version of the game had a lower score in Frustration than the non-adaptive version, although there was no significant difference in Enjoyment. This can be explained by the hardcore players intrinsic characteristics such as the autothelic personality as seen in sections III-B and III-C.

For the casual players, table V shows that the adaptive version (version 1) was more frustrating than the non-adaptive version. We believe this result comes from the fact that the shoot'em up genre is not well familiar for casual players, requiring a specific set of skills such as reflexes and bullet and enemy movement prediction. As Fortugno [21] says, the characteristics that make a game enjoyable and interesting for hardcore players are considered too hard and unencouraging for casual gamers.

Table VI shows for each player what was his/her classification, using C for casual and H for hardcore, his/hers selected initial difficulty for each version (E for Easy, M for Medium and H for Hard), the final difficulty and max difficulty achieved in version 1 (adaptive version), if the player reached the end of the game and the highest wave reached. There were 9 waves, numbered from 0 to 8.

It is clear from table VI that there was a larger number of players that were victorious (that is, players who reached the

TABLE V. COMPARISON OF CEGE SCALES FOR CASUAL PLAYERS

Comparison of CEGE scales for casual players						
		Version 1		Version 2		Difference %
Factors		Sum	Mean	Sum	Mean	
Scale 1	Enjoyment	311	5,759259	336	6,222222	-7,44%
	Frustration	73	2,027778	68	1,888889	7,35%
	CEGE	3145	5,294613	3157	5,314815	-0,38%
	Puppetry	1869	4,944444	1870	4,94709	-0,05%
	Video-game	1276	5,907407	1287	5,958333	-0,85%
Scale 2	Control	923	5,697531	910	5,617284	1,43%
	Facilitators	483	4,472222	489	4,527778	-1,23%
	Ownership	550	4,365079	546	4,333333	0,73%
	Environment	637	5,898148	650	6,018519	-2,00%
	Game-play	639	5,916667	637	5,898148	0,31%

TABLE VI. PLAYER TYPE AND PERFORMANCE

ID	Type	Difficulty									
		Initial		Final			Ended		Max Wave		
		V1	V2	V1	Max	Min	V1	V2	V1	V2	
P	H	E	E	M	M	E	x	x	8	8	
1	C	E	E	E	M	E	x	x	8	8	
2	C	M	M	M	M	M			3	4	
3	C	M	E	E	M	E			4	4	
4	C	E	E	E	E	E			6	4	
5	H	M	M	E	M	E	x	x	8	8	
6	C	E	E	E	E	E			3	4	
7	C	E	E	E	E	E			5	3	
8	H	M	H	M	M	E	x		8	3	
9	H	M	M	M	M	M			4	1	
10	C	E	E	M	M	E			4	4	
11	N	E	E	E	E	E			1	1	
12	C	E	E	E	M	E	x		8	4	
13	H	M	M	E	M	E	x		8	6	
14	C	E	E	E	E	E			1	4	
15	H	M	M	M	M	M	x		8	6	
16	C	M	M	M	M	E			3	6	
17	H	E	E	M	M	E			5	7	
18	C	M	M	M	M	M	x	x	8	8	
19	C	H	H	M	H	M			6	4	
20	H	E	E	E	E	E			6	5	
21	C	E	E	E	E	E			6	5	
22	C	E	E	E	E	E			5	4	
23	H	E	E	E	E	E			5	3	
24	C	E	E	E	E	E			3	5	
25	C	E	E	E	M	E			4	4	
26	H	M	M	E	M	E	x	x	8	8	
27	H	M	M	M	M	M	x	x	8	8	
28	C	M	M	M	M	M	x	x	8	8	
29	H	H	H	M	H	M			8	5	
30	H	E	E	M	M	E	x	x	8	8	
31	H	E	E	E	E	E			7	4	
32	H	E	E	E	E	E			3	3	
33	C	E	E	E	E	E			4	4	
34	H	M	M	M	M	M			3	7	
Total:							12	8			

ending screen) in the adaptive version than in the non-adaptive version (50% greater than the non-adaptive version). Among the these players that finished the game, in 7 of 12 cases (58%) a difference in the difficulty level was observed, whether it was to higher difficulty level or a lower difficulty level.

Post-game interview showed that just 3 players (P, 12, 29) from the 35 total detected actual changes in the difficulty level. Other players (2, 13, 15, 18, 26, 33) said that there was a change in the difficulty level, but the difficulty changes logged in table VI don't show this difference. These and other *non-sequitur* changes some players said they noticed (as changes in graphics or enemy positioning between versions) may be derived from the fact that they could be looking for differences between the versions, as players were told they would be testing two versions of a game.

These result shows that our implementation of dynamic difficulty adjustment was efficient in that it was able to detect the necessity to change the difficulty level and this difference was detected in just 8% of the test cases, and was more favourable to the hardcore player than to the casual player, as expected.

VII. CONCLUSION

The results supported the common-sense idea that hardcore players have a better assimilation of the gaming experience with the adaptive version. This is coherent with the flow theory, as it is expected that hardcore players are more inclined to achieve the flow state as stated in section III-B.

Casual players presented a tendency to prefer the non-adaptive version. One of the possible explanations for this is that shoot'em up games are known to be a niche game for hardcore players and casual players are believed to lack the drive to pursue challenges as defined in section III-C. As Schell [48] says about players and difficulty:

“However, it is the rare player who is persistent enough to win the game, mastering all levels. Most players eventually reach a level where they spend so much time in the frustration zone that they give up on the game.” [48, p. 121].

We consider casual players important for both academia and industry as they are a relatively recently introduced group to gaming (which demands study and represent a potential group for business) and they have great representativity in downloadable games [21, p. 144]. Although it was not possible to demonstrate that in this paper, we believe adaptivity techniques could be used to keep both casual and hardcore players playing a game for longer time by reducing their chance to get into the frustration zone.

As the main contributions of this paper we have: the implementation and case-study of Charles and Black dynamic difficulty adaptivity framework [9], [10]; an efficient implementation of an adaptive shoot'em up game with online learning; the evaluation of dynamic difficulty adaptivity with casual and hardcore players in a shoot'em up game, showing that hardcore players' experience can benefit from the use of dynamic difficulty adaptivity.

ACKNOWLEDGMENTS

This work was partially supported by CNPq (National Council for Scientific and Technological Development) and FINEP (Brazilian Innovation Agency), both linked to the Ministry of Science, Technology, and Innovation, CAPES (Coordination for the Improvement of Higher Education Personnel, linked to the Ministry of Education), and the Department of Informatics/PUC-Rio.

REFERENCES

- [1] G. Andrade, G. Ramalho, A. S. Gomes, and V. Corruble, “Dynamic game balancing: An evaluation of user satisfaction,” in *AAAI conference on Artificial Intelligence and Interactive Digital Entertainment*, 2006, p. 38.
- [2] R. Android, “GundeadliGne,” Japan, Jun. 2010.
- [3] S. C. Bakkes, P. H. Spronck, and H. Jaap van den Herik, “Opponent modelling for case-based adaptive game AI,” *Entertainment Computing*, vol. 1, no. 1, p. 2737, 2009.
- [4] R. A. Bartle, “Players who suit MUDs,” *Journal of Online Environments*, vol. 1, no. 1, 1996. [Online]. Available: <http://www.mud.co.uk/richard/hclds.htm>
- [5] R. Bernhaupt, “User experience evaluation in entertainment,” in *Evaluating User Experience in Games: Concepts and Methods*, 1st ed., R. Bernhaupt, Ed. London: Springer, 2010, pp. 3–7.
- [6] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhardt, and J. N. Pidruzny, “The development of the game engagement questionnaire: A measure of engagement in video game-playing,” *Journal of Experimental Social Psychology*, vol. 45, no. 4, pp. 624–634, Jul. 2009. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0022103109000444>
- [7] E. H. Cálvillo-Gómez, “On the core elements of the experience of playing video games,” Ph.D. dissertation, UCL (University College London), 2009. [Online]. Available: <http://discovery.ucl.ac.uk/18510/>
- [8] E. Cálvillo-Gómez, P. Cairns, and A. Cox, “Assessing the core elements of gaming experience,” in *Evaluating User Experience in Games: Concepts and Methods*, 1st ed., R. Bernhaupt, Ed. London: Springer, 2010, pp. 47–71.
- [9] D. Charles and M. Black, “Dynamic player modeling: A framework for player-centered digital games,” in *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, p. 2935.
- [10] D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kcklich, A. Moore, and K. Stringer, “Player-centred game design: Player modelling and adaptive digital games,” in *Proceedings of the Digital Games Research Conference*, vol. 285, 2005.
- [11] J. Chen and T. Fullerton, “Flow in games,” Master’s thesis, University of Southern California, California, USA, 2006. [Online]. Available: http://www.jenovachen.com/flowingames/Flow_in_games_final.pdf
- [12] Compile, “Zanac,” Japan, Nov. 1986.
- [13] —, *Zanac Instruction Manual*. USA: Fujisankei Communications International, 1986.
- [14] B. Cowley, D. Charles, M. Black, and R. Hickey, “Toward an understanding of flow in video games,” *Computers in Entertainment*, vol. 6, no. 2, p. 1, Jul. 2008. [Online]. Available: <http://portal.acm.org/citation.cfm?doi=1371216.1371223>
- [15] C. Crawford, *The art of computer game design*. Osborne/McGraw-Hill Berkley, 1984.
- [16] M. Csikszentmihalyi, *Flow : the psychology of optimal experience*, 1st ed. New York: Harper & Row, 1990.
- [17] P. Demasi and A. J. d. O. Cruz, “Aprendizado de regras nebulosas em tempo real para jogos eletrônicos,” in *XI Simpósio Brasileiro de Sistemas Multimídia e Web. WJogos - II Workshop Brasileiro de Jogos e Entretenimento Digital*, Salvador, Bahia, Brazil, Nov. 2003. [Online]. Available: http://www.labic.nce.ufrj.br/downloads/2wbjed_wjogos_2003_1.pdf
- [18] —, “Evolução de agentes em tempo real para jogos eletrônicos de ação,” in *XI Simpósio Brasileiro de Sistemas Multimídia e Web. WJogos - II Workshop Brasileiro de Jogos e Entretenimento Digital, Salvador - BA, 4 e 5 de novembro de 2003.*, Salvador, Bahia, Brazil, Nov. 2003. [Online]. Available: http://www.labic.nce.ufrj.br/downloads/2wbjed_wjogos_2003_2.pdf
- [19] —, “Estratégias adaptativas e evolutivas em tempo real para jogos eletrônicos,” Master’s thesis, UFRJ, Rio de Janeiro, RJ, Brasil, 2003.
- [20] R. Entertainment, “Max payne,” Jul. 2001.
- [21] N. Fortugno, “The strange case of the casual gamer,” in *Game Usability: Advice from the experts for advancing the player experience*, 1st ed., K. Isbister and N. Schaffer, Eds. Massachusetts, USA: Morgan Kaufman, 2008, pp. 143–158.
- [22] T. Fullerton, C. Swain, and S. Hoffman, *Game Design Workshop : A playcentric approach to creating innovative games*. Amsterdam, Boston: Elsevier Morgan Kaufmann, 2008.
- [23] K. M. Gilleade and A. Dix, “Using frustration in the design of adaptive videogames,” in *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 2004, p. 228232.
- [24] H. Hoonhout, “Let the game tester do the talking: Think aloud and interviewing to learn about the game experience,” in *Game Usability: Advice from the experts for advancing the player experience*, 1st ed., K. Isbister and N. Schaffer, Eds. Massachusetts, USA: Morgan Kaufman, 2008, pp. 65–77.
- [25] R. Houlette, “Player modeling for adaptive games,” in *AI Game Programming Wisdom 2*, 1st ed., S. Rabin, Ed. Boston, MA: Charles River Media, 2004, pp. 557–566.
- [26] J. Huizinga, *Homo Ludens: o jogo como elemento da cultura*, 6th ed., ser. Estudos. São Paulo: Perspectiva, 2010.
- [27] R. Hunicke, “The case for dynamic difficulty adjustment in games,” in *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, vol. 256. ACM, 2005, pp. 429 – 433.
- [28] R. Hunicke and V. Chapman, “AI for dynamic difficulty adjustment in games,” in *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, 2004, p. 9196.

- [29] J. Ibañez and C. Delgado-Mata, “Adaptive two-player videogames,” *Expert Systems with Applications*, vol. 38, pp. 9157–9163, Aug. 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0957417411001151>
- [30] R. Ierusalimschy, *Programming in Lua*, 2nd ed. Rio de Janeiro: Lua.org, 2006.
- [31] J. Juul, “The game, the player, the world: Looking for a heart of gameness,” *Level Up: Digital Games Research Conference Proceedings*, pp. 30–45, 2003.
- [32] R. Koster, *A Theory of Fun for Game Design*, 1st ed. Scottsdale AZ: Paraglyph Press, 2004.
- [33] M. Lankes, R. Bernhaupt, and M. Tscheligi, “Evaluating user experience factors using experiments: Expressive artificial faces embedded in contexts,” in *Evaluating User Experience in Games: Concepts and Methods*, 1st ed., R. Bernhaupt, Ed. London: Springer, 2010, pp. 165–183.
- [34] R. Lopes and R. Bidarra, “Adaptivity challenges in games and simulations: A survey,” *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 2, pp. 85–99, Jun. 2011. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5765665>
- [35] A. F. d. V. Machado, C. G. Duarte, H. E. Rogério, L. M. d. Paulo, W. C. S. d. Paula, E. W. G. Clua, and T. J. Rogers, “Dynamic difficulty balancing of strategy games through player adaptation using top culling,” in *Proceedings of SBGames 2011*, Salvador, Bahia, Brazil, Nov. 2011.
- [36] M. Machado, E. Fantini, and L. Chaimowicz, “Player modeling: Towards a common taxonomy,” in *Computer Games (CGAMES), 2011 16th International Conference on*, 2011, p. 5057.
- [37] —, “Player modeling: What is it? how to do it?” in *Proceedings of SBGames 2011*, Salvador, Bahia, Brazil, Nov. 2011.
- [38] O. Missura and T. Grtner, “Player modeling for intelligent difficulty adjustment,” in *Discovery Science*, 2009, p. 197211.
- [39] —, “Predicting dynamic difficulty,” in *Proceedings of the Ninth Workshop on Mining and Learning with Graphs*, San Diego, CA, 2011.
- [40] Nintendo EAD, “Mario kart 64,” Dec. 1996.
- [41] A. Noon, A. Philippides, and A. Thompson, “Dynamic difficulty adjustment in games: A neuroevolutionary approach,” Master’s thesis, University of Sussex, East Sussex, England, Sep. 2009.
- [42] J. Novak, *Desenvolvimento de Games*. Brazil: Cengage Learning, 2011.
- [43] C. Pedersen, J. Togelius, and G. Yannakakis, “Modeling player experience in super mario bros,” in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 2009, p. 132139.
- [44] S. Poole, *Trigger Happy: Videogames and the entertainment revolution*. Publicado online pelo autor, 2007. [Online]. Available: <http://stevenpoole.net/trigger-happy/>
- [45] J. Ranhel, “O conceito de jogo e os jogos computacionais,” in *Mapa do Jogo: A diversidade cultural dos games*, 1st ed., ser. Profissional, L. Santaella and M. Feitoza, Eds. São Paulo: Cengage Learning, 2009, pp. 3–22.
- [46] K. Salen and E. Zimmerman, *Rules of play : game design fundamentals*. Cambridge Mass.: MIT Press, 2003.
- [47] C. Santos and F. Osrio, “Applying the AdapTIVE model in games,” in *Anais WJogos 2005, SBGames 2005*, M. Zuffo, R. Lopes, and I. Souza, Eds., So Paulo, SP, 2005, pp. 264–269.
- [48] J. Schell, *A Arte de Game Design: o livro original*, 1st ed., ser. Media Technology. Elsevier, 2011.
- [49] A. Smith, C. Lewis, K. Hullett, G. Smith, and A. Sullivan, “An inclusive taxonomy of player modeling,” Center for Games and Playable Media, University of California, Santa Cruz, Santa Cruz, CA, Tech. Rep. UCSC-SOE-11-13, 2011.
- [50] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, “Adaptive game AI with dynamic scripting,” *Machine Learning*, vol. 63, no. 3, pp. 217–248, Mar. 2006. [Online]. Available: <http://www.springerlink.com/index/10.1007/s10994-006-6205-6>
- [51] B. G. Studios, “The elder scrolls IV: oblivion,” Mar. 2006.
- [52] thatgamecompany, “flOw,” California, USA, Apr. 2006.
- [53] Valve Corporation, “Left 4 dead,” Nov. 2008.
- [54] —, “Left 4 dead 2,” Nov. 2009.
- [55] H. van den Herik, H. Donkers, and P. Spronck, “Opponent modelling and commercial games,” *Proceedings of IEEE*, p. 1525, 2005.
- [56] B. Weber, “Infinite adaptive mario,” 2010. [Online]. Available: http://users.soe.ucsc.edu/~bweber/dokuwiki/doku.php?id=infinite_adaptive_mario
- [57] B. Widrow and M. E. Hoff, “Adaptive switching circuits,” in *Neurocomputing: foundations of research*, J. A. Anderson and E. Rosenfeld, Eds. Cambridge, MA, USA: MIT Press, 1988, p. 123134. [Online]. Available: <http://dl.acm.org/citation.cfm?id=65669.104390>
- [58] G. Xavier, *A condição eletrolúdica: Cultura visual nos jogos eletrônicos*. Teresópolis, Brasil: 2AB Editora, 2010.
- [59] G. N. Yannakakis, “How to model and augment player satisfaction: A review,” in *Proceedings of the 1st Workshop on Child, Computer and Interaction, Chania, Crete, ACM Press*, 2008.
- [60] G. N. Yannakakis and J. Hallam, “Real-time adaptation of augmented-reality games for optimizing player satisfaction,” in *Computational Intelligence and Games, 2008. CIG’08. IEEE Symposium On*, 2008, p. 103110.
- [61] G. Yannakakis and M. Maragoudakis, “Player modeling impact on player’s entertainment in computer games,” in *User modeling 2005: 10th international conference*, ser. Lecture notes in computer science, L. Ardissono, P. Brna, and A. Mitrovic, Eds., vol. 3538. Edingburgh, Scotland, UK: Springer, Jul. 2005, pp. 74–78.