Finding Comfortable Settings of Video Games: Using Game Refinement Measure in Pokemon Battle AI

Chetprayoon Panumate[,a]   Hiroyuki Iida[,b]

Abstract: This paper explores an innovative way to find a comfortable setting of video games. Pokemon is chosen as a benchmark and game refinement theory is employed for the assessment. Pokemon has many entertaining factors to be considered like other video games. Interestingly, there are some factors which are still remained after the first episode of Pokemon was released in 1996. Among them, the number of Pokemon that one trainer can carry ($n = 6$) has never been changed. It is assumed that the setting may be optimal, which has been realized by players as well as composers. In this paper, Pokemon battle game is simulated by simplifying many factors and different types of computer players (AIs) are developed for the experiments. The experimental results show that the setting ($n = 6$) is the best in the sense that the setting covers most players of various levels and that Pokemon is mostly played by children. Moreover, there are other reasonable settings for some specific levels. For example, $n = 5$ and $n = 7$ may be a comfortable setting for experts and beginners, respectively.

Keywords: Pokemon, Pokemon AI, Entertainment, Game refinement theory, Setting of video games

1. Introduction

Every game would have its own suitable setting to maximize the entertaining impact, which makes the game different from others. However, it is a big challenge to identify such a suitable setting. For example, the rules of chess have been elaborated with many changes in its evolutionary history [1]. We know through our experience in developing game systems that a setting is decided by some hidden mechanisms to make the game survive for many decades or to be more successful in the market.

Nowadays, there are many video games. Some are so popular but others are not so. There are many reasons behind this fact, e.g., its story, graphics etc. In a practical sense, one of the most important points is the setting of some variables in the game. Its setting may often considerably affect the entertaining aspect which leads the game to be exciting or boring. That is the reason why we believe that each setting of video game has its own reasons behind this, and therefore finding a comfortable setting can be significantly important to maximize the entertainment in playing the game under consideration. Thus, this paper proposes an innovative approach to find a comfortable setting of video games.

In this study, we use the game of Pokemon as shown in Figure 1, one of the most popular video games [2]. While many efforts have been devoted to the study of Pokemon with a focus on different aspects such as education [3], media science [4], social science [5] and computer science [6], the present study starts from the foundational question: "Why has Pokemon been popular so long time?"

So, we focus on Pokemon battle which is the core part of of

Fig. 1: A screenshot of Pokemon Emerald

Pokemon game. Actually, there are many factors to be considered in Pokemon battle. Some of them have been added, deleted or changed in order to improve the quality of Pokemon battle such as abilities, some moves, the number of types and so on. However, a few factors are still same after the first episode of Pokemon was released in 1996. Among them, the number of Pokemon ($n = 6$) that one trainer can carry has never been changed.

Beside this, game theory [7] originated with the idea of the existence of mixed-strategy equilibrium in two-person zero sum games. It has been widely applied as a powerful tool in many fields such as economics, political science and computer science. Game refinement theory [8] [9] is another game theory focusing on attractiveness and sophistication of games based on uncertainty of game outcome. Classical game theory concerns the winning strategy from the players point of view, whereas game refinement theory concerns the entertaining impact from the game designers point of view.

In this study we use a measure which is derived from the game refinement theory to find comfortable settings of Pokemon game. We simulate Pokemon battles and its AIs in order to collect the data. However, it is hard to create a perfect human-like AI. Instead, we create different types of AIs to estimate the real human's behaviour. We then figure out a reasonable model of game refinement for Pokemon game.

The structure of this paper is as follows. We first briefly introduce Pokemon and explain how to simulate the game and its AIs in Section 2. Then, in Section 3, we present the fundamental idea of game refinement theory and describe how to apply game refinement theory to Pokemon battle. Finally, the experiments are performed and the results obtained are discussed in Section 4. Finally, concluding remarks are given in Section 5.

2. Pokemon Battle and AI

This section presents Pokemon-related matters such as its overview, outline of Pokemon battle and its battle simulator and our Pokemon AIs.

2.1 Overview of Pokemon

Pokemon [10] is a series of games developed by Game Freak and Creatures Inc [*1]. First released in 1996 in Japan for the Game Boy, the main series of role-playing video games (RPG) has continued on each generation of handheld Nintendo consoles, including Game Boy Color, Game Boy Advance, Nintendo DS and Nintendo 3DS [11]. Pokemon is the most successful computer game ever made, the top globally selling trading-card game of all time and one of the most successful children's television programs ever broadcast [12]. While the main series consists of role-playing games, spinoffs encompass other genres such as action role-playing, puzzle and digital pet games.

2.2 Pokemon battle

The goal of Pokemon game is to win the badges of gyms and become the champion of the league [13]. For this purpose, one has to win every battle in the game. In this study, we focus on each battle. The goal of Pokemon battle is to fight until the opponent has no Pokemon remained. One Pokemon will be fainted if its HP (hit points or health points) reaches zero [14].

Normally, each player will be called as a trainer because the duty of a player is to train his or her Pokemon to be powerful enough to clear the game. Each trainer can bring up six Pokemons and each Pokemon can remember four moves. Each Pokemon has six kinds of stats composed of HP, attack, defense, special attack, special defense and speed. There are eighteen types of Pokemon and one Pokemon can be only one or two types such as Pikachu which is an electric type or Golem which is both rock type and ground type. The eighteen types consist of bug, dark, dragon, electric, fairy, fight, fire, flying, ghost, grass, ground, ice, normal, poison, psychic, rock, steel and water. The effectiveness of each move depends on the type of move and type of Pokemon that received move. Some examples are shown in Table 1.

[*1] All products, company names, brand names, trademarks, and sprites are properties of their respective owners. Sprites are used here under Fair Use for the educational purpose.

Table 1: Examples of types and its effectiveness

		Defending Type			
Atta-cking Type	Type	Fire	Grass	Ground	Water
	Fire	×0.5	×2	×0.5	×0.5
	Grass	×0.5	×0.5	×2	×2
	Ground	×2	×0.5		
	Water	×2	×0.5	×2	×0.5

Normally, a trainer can choose what he/she wants to do among four choices: Fight, Switching Pokemon, Using Items and Run as shown in Figure 1. However, in Pokemon battle rules, trainer cannot choose run or using item so trainer can choose only fight or changing Pokemon. There are many kinds of moves. It varies from offensive attacks like Tackle to defensive moves like Protect. Some attack moves are physical attack which will use one's attack stat and the opponent's defense stat to be calculated the results such as Slam while some attack moves are special attack which will use one's special attack stat and the opponent's special defense stat to be calculated the results such as Psybeam. Some moves can change one's stat and/or the opponent's stats such as Tail Whip and Growl. Some moves inflict status effects to the target like paralyzed and poison such as Thunder Wave and Toxic.

Turn's order is decided by the speed of the two Pokemons which are currently fighting. The fastest one simply goes first. Switching Pokemon always goes before attacks. Moreover, each Pokemon has special ability which will effect in Pokemon battle such as Sniper ability that let the critical hit more powerful. Each Pokemon can hold one item that makes holder more forceful such as Leftovers which make holder regains its HP every turn. The trainer is the winner if he or she has remained Pokemon but the opponent has no Pokemon.

2.3 Pokemon battle simulator

One possible way to collect many data of Pokemon battle which have been performed by human players is given by online Pokemon battle simulator [15]. These data have been used to analyze the entertainment impact with a focus on battle [16]. However, in this study we need to change the number of Pokemons that one trainer can carry in order to find a comfortable setting. Therefore, we simulate the Pokemon battle game by simplifying many factors in the game.

First, we introduce some equations from [17] and [18] which are essential to create a simulated Pokemon battle game. When we generate a Pokemon, its stat needs to be generated. Since one Pokemon is composed of six stats, its HP stat can be defined by Equation (1).

$$HP = \left\lfloor \frac{(2 \times Base + IV + \lfloor \frac{EV}{4} \rfloor) \times Lv}{100} \right\rfloor + Lv + 10 \qquad (1)$$

Where IV stands for an individual value which is randomly generated by the game at the time when one meets that Pokemon first. There are six IV because each Pokemon has six battle stats. $Base$ means the initial value of that stat. There are six $Base$ because each Pokemon has six battle stats. To calculate HP, IV of HP and $Base$ of HP are used. For stats of Pokemon considered, it

depends on the kind of ones Pokemon. Some Pokemons have outstanding *Base* of HP while other Pokemons have poor *Base* of HP. Rationally, Pokemon with poor *Base* of HP should have another great initial stat. However, legendary Pokemon may have excellent value in every initial stat. *EV* stands for a special value which Pokemon will receive after the battle. It depends on the kind of Pokemon which has been defeated. Some Pokemons give *EV* of HP while other Pokemons give *EV* of other stats. *Lv* denotes a level of Pokemon considered. It simply starts from 1 to 100. The more Pokemons level increases, the stronger Pokemon becomes.

Equation (2) shows how to calculate *OS*, which means other stats. Simply, other stats are calculated by using *IV*, *EV* and *Base* of the stat considered.

$$OS = \left\lfloor \left(\left\lfloor \frac{(2 \times Base + IV + \lfloor \frac{EV}{4} \rfloor) \times Lv}{100} \right\rfloor + 5 \right) \times Nat \right\rfloor \quad (2)$$

Where *Nat* is the nature of the Pokemon considered. It is the mechanic that influences how a Pokemon's stats grow. There are 25 natures composed of hardy, lonely, brave, adamant, naughty, bold, docile, relaxed, impish, lax, timid, hasty, serious, jolly, naive, modest, mild, quiet, bashful, rash, calm, gentle, sassy, careful and quirky. Each nature has its own property. For example, adamant nature will increase the attack stat while reducing the special attack stat.

Equation (3) shows how to calculate damage.

$$Damage = \left(\frac{2 \times Lv + 10}{250} \times \frac{Atk}{Def} \times Base + 2 \right) \times Mod \quad (3)$$

Where *Lv* is the level of the attacking Pokemon. *Atk* and *Def* are the working Attack and Defense stats of the attacking and defending Pokemon, respectively. If the attack is Special, the Special Attack and Special Defense stats are used instead. *Base* is the base power of the attack. *Mod* can be defined by Equation (4).

$$Mod = STAB \times Type \times Critical \times Other \times Random \quad (4)$$

Where *STAB* is the same-type attack bonus. It is equal to 1.5 if the attack is the same type as the user, and 1 otherwise. *Type* is the type effectiveness. This can be either 0, 0.25, 0.5, 1, 2, or 4 that is determined depending on the type of attack and the type of the defending Pokemon. *Critical* is 1.5 if this attack is a critical attack, and 1 otherwise. *Other* stands for the things like held items, abilities, field advantages, and whether the battle is a Double Battle or Triple Battle or not. *Random* is a random number from 0.85 to 1.00.

However, some complex factors such as holding items component, abilities component and some complex moves are not considered. Finally, we simulate Pokemon battle game. We implement 20 of the most used Pokemons from the total number of 718 Pokemons [19]. We also implement its most used move according to reliable site [20].

2.4 Pokemon battle AI

For the experiments we propose four different types of Pokemon battle AI: Random AI, Attack AI, Smart-Attack AI and Smart-Defense AI. The details are given below.

Algorithm 1 Random AI

1: **procedure** RANDOMAIDECISION
2:    $n_1 \leftarrow number\ of\ Moves$
3:    $n_2 \leftarrow number\ of\ remainning\ Pokemons$
4:    $r \leftarrow random\ from\ 0\ to\ (n_1 + n_2 - 1)$
5:    **if** $r < n_1$ **then**
6:       **return** $useMove(r)$
7:    **else**
8:       **return** $changePokemonTo(r - n_1 + 1)$
9:    **end if**
10: **end procedure**

Algorithm 2 Attack AI

1: **procedure** ATTACKAIDECISION
2:    $n \leftarrow list\ of\ Moves$
3:    $dmgTemp1 \leftarrow calDmg(n_0, opponent)$
4:    $moveIndex \leftarrow 0$
5:    **for** each n_i in n **do**
6:       $dmgTemp2 \leftarrow calDmg(n_i)$
7:       **if** $dmgTemp2 > dmgTemp1$ **then**
8:          $dmgTemp1 \leftarrow dmgTemp2$
9:          $moveIndex \leftarrow i$
10:       **end if**
11:    **end for**
12:    **return** $useMove(moveIndex)$
13: **end procedure**

2.4.1 Random AI

Random AI will choose at random every possible choice including the move and the change of Pokemons. This algorithm is simply described in Algorithm 1.

2.4.2 Attack AI

Attack AI will choose a move that makes the highest damage from the currently used Pokemon. It will not change the currently used Pokemon. This algorithm is described in Algorithm 2.

2.4.3 Smart AI

We design Smart AI with the purpose to create a simple human-like Pokemon battle AI. We create two types of Smart AIs: Smart-Attack AI and Smart-Defense AI. These algorithms are described in Algorithm 3.

Smart-Attack AI checks at first whether or not the currently used Pokemon has a move that wins the opponent's type. If yes, it selects the best move that makes the highest damage from the moves that currently used Pokemon has. If not, it checks whether or not it has other Pokemons that have a move that wins the opponent's type. If yes, it changes the currently used Pokemon with the Pokemon that has a move that makes the highest damage to the opponent. If not, it has to select the move that makes the highest damage from the currently used Pokemon inevitably. Therefore, its $checkType(n_0, opponent)$ can be denoted as $hasMoveWinType(n_0, opponent)$ which means that n_0 has a move that wins the *opponent*'s type.

Smart-Defense AI checks at first whether or not the currently used Pokemon's type wins the opponent's type. If yes, it selects the best move that makes the highest damage from the currently

Algorithm 3 Smart-Attack AI and Smart-Defense AI

```
 1: procedure SMARTAIDECISION
 2:     n ← list of Pokemons
 3:     opponent ← opponent's current Pokemon
 4:     moveIndex ← findMaxDmgMove()
 5:     if checkType(n_0, opponent) then
 6:         return useMove(moveIndex)
 7:     else
 8:         changePokemon ← false
 9:         dmgTemp ← 0
10:         pokemonIndex ← 1
11:         for each n_i in n do
12:             if checkType(n_i, opponent) then
13:                 changePokemon ← true
14:                 dmgTemp2 ← findMaxDmg()
15:                 if dmgTemp2 > dmgTemp then
16:                     dmgTemp ← dmgTemp2
17:                     pokemonIndex ← i
18:                 end if
19:             end if
20:         end for
21:         if changePokemon then
22:             return changePokemon(pokemonIndex)
23:         else
24:             return useMove(moveIndex)
25:         end if
26:     end if
27: end procedure
```

used Pokemon same as Smart-Attack AI. If not, it checks whether or not it has other Pokemons whose type wins the opponent's type. If yes, it changes the currently used Pokemon with the Pokemon whose type wins the opponent's type and has a move that makes the highest damage to the opponent. If not, it has to select the move that makes the highest damage from the currently used Pokemon inevitably. Therefore, its $checkType(n_0, opponent)$ can be denoted as $winType(n_0, opponent)$ which means that n_0's type wins $opponent$'s type.

The significant difference between Smart-Attack AI and Smart-Defense AI is that Smart-Attack AI checks type from its moves while Smart-Defense AI checks type from Pokemon. So, Smart-Attack AI would make higher damage because the damage is based on the move. However, Smart-Defense AI is safer because if your Pokemon's type wins opponent's Pokemon's type, it is hard that your opponent can make high damage to you. This significant difference leads Smart-Attack AI can end the game faster than Smart-Defense AI.

3. Assessment of Pokemon Battle

This section first gives a short sketch of game refinement theory. We then discuss on how to apply it to Pokemon battle game. Using game refinement measures, Pokemon battles with various number of Pokemons are compared.

3.1 Game refinement theory

A general model of game refinement was proposed based on the concept of game progress and game information progress [9]. It bridges a gap between board games and sports games. The 'game progress' is twofold. One is game speed or scoring rate, while another one is game information progress with focus on the game outcome. Game information progress presents the degree of certainty of game's results in time or in steps.

Having full information of the game progress, i.e. after its conclusion, game progress $x(t)$ will be given as a linear function of time t with $0 \leq t \leq t_k$ and $0 \leq x(t) \leq x(t_k)$, as shown in Equation (5).

$$x(t) = \frac{x(t_k)}{t_k} t \tag{5}$$

However, the game information progress given by Equation (5) is unknown during the in-game period. The presence of uncertainty during the game, often until the final moments of a game, reasonably renders game progress as exponential. Hence, a realistic model of game information progress is given by Equation (6).

$$x(t) = x(t_k)(\frac{t}{t_k})^n \tag{6}$$

Here n stands for a constant parameter which is given based on the perspective of an observer of the game considered. Then acceleration of game information progress is obtained by deriving Equation (6) twice. Solving it at $t = t_k$, we have Equation (7).

$$x''(t_k) = \frac{x(t_k)}{(t_k)^n} t^{n-2} n(n-1) = \frac{x(t_k)}{(t_k)^2} n(n-1) \tag{7}$$

It is assumed in the current model that game information progress in any type of game is encoded and transported in our brains. We do not yet know about the physics of information in the brain, but it is likely that the acceleration of information progress is subject to the forces and laws of physics.

Therefore, we expect that the larger the value $\frac{x(t_k)}{(t_k)^2}$ is, the more the game becomes exciting, due in part to the uncertainty of game outcome. Thus, we use its root square, $\frac{\sqrt{x(t_k)}}{t_k}$, as a game refinement measure for the game under consideration. We call it R value for short as shown in Equation (8).

$$R = \frac{\sqrt{x(t_k)}}{t_k} \tag{8}$$

3.2 Board games approach

Game refinement values for various board games such as chess, shogi, Go and Mah-Jong were calculate in [8]. Let B and D be the average branching factor (number of possible options) and average game length (depth of whole game tree), respectively. Therefore, in general, R value for board game is obtained by Equation (9) and game refinement values of major board games are shown in Table 2.

$$R = \frac{\sqrt{B}}{D} \tag{9}$$

3.3 Application to Pokemon battle

Pokemon battle is a turn-based game where each player has to choose what to do at his/her turn, same as board games like chess. So we can suppose Pokemon battle as a board game [16].

Table 2: Measures of game refinement for major board games

Game	B	D	R
Western chess	35	80	0.074
Chinense chess	38	95	0.065
Japanese chess	80	115	0.078
Go	250	208	0.076
Mah Jong	10.36	49.36	0.078

However, there are many important differences between Pokemon battle and board game, as described below.

One-to-one fighting

Pokemon battle is one-to-one fighting. This means that each player can control only one Pokemon to fight with the opponent's Pokemon. If one wants to use another Pokemon, the player has to change his/her current used Pokemon [*2] Chess, a typical board game, is a kind of war with a lot of pieces in the battlefield. Player can choose which piece that they want to control in his/her turn. Thus, chess is run by many pieces against many pieces of the opponent.

Position

Chess has a position on a board at each turn. This is a very important factor for board game. That is why we call 'board' game. But in Pokemon battle, there is no such position. A game is normally run by one Pokemon per player, i.e., one's Pokemon and opponent's Pokemon. They play what they want to do, whereas no positions are considered.

Turn's sequence and turn's meaning

Turn in chess means the chance that one can do what the player wants. Turn is run by switching system. $A's\ turn \rightarrow B's\ turn \rightarrow A's\ turn \rightarrow B's\ turn \rightarrow ...$, simple like this. In Pokemon battle, one turn means that both players have to choose what they want to do. Then, the speed (one of Pokemon's stats) of current used Pokemon is the factor which is decided on who will first start this turn. Pokemon with a faster speed can do first in that turn, then Pokemon with a lower speed will do afterward. To summarize the meaning of turn, one turn in chess is for only one side player's action (one ply), whereas Pokemon battle's one turn means simultaneous actions of both players. As for the turn's sequence, it is a simple order in chess but it is ordered in Pokemon battle by speed of current used Pokemon.

Hence, game-refinement theory can be applied by using the same idea as board game. Normally, we find R value by using the average possible options (say B) and game length (say D), as shown in Equation (9). For possible options B, in Pokemon battle, they are very limited because one Pokemon can remember only four moves and player can change his/her current used Pokemon to other Pokemons that player possesses. In case of using some items such as medicine and potion, it is illegal in Pokemon contests, so we do not consider that case. For example, basically, one can carry at most six Pokemons, so player can choose five possible options when player wants to change the current used Pokemon. Therefore, we may use possible options equal to 9 which comes from four moves and five remained Pokemon. However, if the

---

[*2] There are two-to-two or three-to-three Pokemon fighting mode but not so popular.

time passes, the number of Pokemon that player can change will be reduced because their Pokemons are fainted due to the battle. Also, the possible options will be reduced. Therefore, we can find an average possible options by the summation of possible options of each turn divided by the number of turns.

For game length D, the meaning of one turn in Pokemon battle is both players' simultaneous actions. Hence, there are two actions in one turn. So, we have to multiply 2 to this value. Finally, with this simple method, we can completely find R value as shown in Equation (10).

$$R = \frac{\sqrt{B}}{D} = \frac{\sqrt{Average\_number\_of\_options}}{Average\_number\_of\_turns \times 2} \quad (10)$$

4. Results and Discussion

Experiments are performed by adjusting the number of Pokemons that one trainer can carry, denoted by n. We use AI in our simulated Pokemon battle game in four algorithms, described in Section 2.4. For each n in each type of AI, we perform one million games. Then, we collect B and D. We then calculate game refinement values by using Equation (10) and the results are shown in Table 3.

Table 3: Measures of game refinement for Pokemon battle

n	Random AI	Attack AI	S-Atk AI	S-Dfs AI
1	0.222	0.296	0.296	0.296
2	0.124	0.173	0.169	0.160
3	0.083	0.127	0.119	0.107
4	0.060	0.103	0.093	0.079
5	0.046	0.088	0.076	0.063
6	0.037	0.077	0.065	0.052
7	0.031	0.069	0.057	0.044
8	0.026	0.063	0.051	0.039
9	0.022	0.059	0.046	0.035
10	0.020	0.055	0.043	0.032

4.1 Game refinement values and player's level

For Random AI, due to its random algorithm, there are many times that it selects changing Pokemon or selects some bad moves which are not effective. These decisions make the game end slower than expected. Actually, real human player would select a reasonable choice which makes the game end faster than Random AI. In contrast, Attack AI would select the best choice which maximizes the damage without changing Pokemon and it leads the game to end very fast. Practically, real human players should not be able to end the game as fast as Attack AI can. Also, human players should not be able to end the game as slow as Random AI does. Therefore, we can simply conclude that these two kinds of AI can show the possible range of R value based on player's skill. Random AI and Attack AI will be a lower and upper bound, respectively.

To go deeply, we simulate two types of Smart AI: Smart-Attack AI and Smart-Defense AI. It is expected that the two AIs show human-like performance. We can roughly classify Pokemon's player in many levels such as novice, beginner, normal and expert.

Novice players would play like Random AI due to the very poor knowledge, so they do not know how to choose a good move. Beginner players would play like Attack AI because they try to end the game as fast as possible by choosing the maximum damage move which is not hard to calculate. Normal player would play like Smart AI by using simple decision making as we describe. However, higher level players likes expert players would know that Pokemon battle is more complex than other levels think. So, these level players would choose many complex moves which make the game somehow long but not so long as Random AI does. We show, in Figure 2, the relation between R-value and performance level for various number of Pokemons that a trainer can carry.
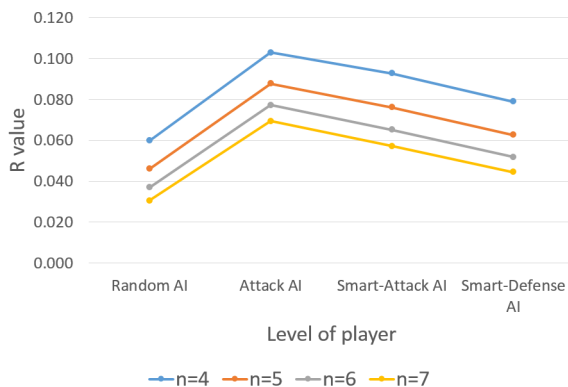

Fig. 2: R values and performance level for various number of Pokemons (n)

4.2 Summary of the experimental results

According to our previous studies, it is found that many sophisticated games have R value between $0.07 - 0.08$ which may be suitable for normal viewers. It suggests that for the Pokemon battle $n = 5$ may be the best setting. On the other hands, it is observed in [16] that the main target of Pokemon game is children. So its R value should be slightly lower than the sophisticated zone in order to have the appropriate level of excitement. Hence it is reasonably assumed that the appropriate R value for Pokemon battle is around $0.06 - 0.07$. Under the assumption, $n = 6$ is the best setting.

Moreover, $n = 5$ and $n = 7$ are also reasonable settings for some specific levels of Pokemon player. For example, if we focus on beginner's level, $n = 7$ may be nice. If we focus on expert's level, $n = 5$ may be fine. However, when focusing on children's performance, we see that $n = 6$ has the best R value because the range is very comprehensive. It can hold an acceptable value for novice, beginner, normal and expert. This is the answer to the research question in this study: Why the number of Pokemons that one trainer can carry is six? Because it is an appropriate number which is optimal for every level of players. This is the reason why the number six has never been changed after the first episode of Pokemon was released in 1996.

Our approach to find comfortable settings of Pokemon can be summarized as follows. We first simulate a Pokemon battle game by simplifying many factors. The simulated game should be as simple as possible, but it should be sufficiently complex in order to create a realistic simulated game. Then, we try to implement some realistic AIs for gathering data. However, it is hard to make a perfect human-like AI. Therefore, our solution is that we create some AIs which can be a lower bound and the upper bound of the human performance. Then, we try to create a human-like AI as realistic as possible. We can estimate the real human's performance based on his level by considering these AIs. Finally, a reasonable game refinement model is applied and we perform our experiment by the numerous games.

5. Conclusion

This paper proposed an innovative way to find the comfortable settings in video games. Pokemon was chosen as a benchmark and game refinement measure was employed for the assessment for which statistical data of Pokemon battle games were collected. Pokemon battle AIs of four different levels were developed and experiments were performed. From the experimental results we arrived at a conclusion below. In the Pokemon battle game the setting ($n = 6$) is most comfortable under the assumption that Pokemon is mostly played by children and comfortable zone for children is somewhere between 0.06 to 0.07. Moreover, there are other reasonable settings for some specific levels. For example, $n = 5$ and $n = 7$ may be a comfortable setting for experts and beginners, respectively.

It is understood that the work presented here is a simple model with no complicated factors and more studies are required. This work has some limitations such as the number of Pokemons implemented, Pokemon battle's AI and some Pokemon battle mechanics which is hard to implement are ignored. Further works may improve these points. Also, we can raise another research question such as why the number of moves (say m) that one Pokemon can remember must be four. It is expected to apply our approach to justify the original settings $m = 4$. Moreover, investigation in various domains such as Pokemon contest or another video game can be important to improve our approach.

References

[1] A. Cincotti, H. Iida, and J. Yoshimura, "Refinement and complexity in the evolution of chess," in *the 10th International Conference on Computer Science and Informatics*. Springer, 2007, pp. 650–654.

[2] J. Horton, "got my shoes, got my pokémon: Everyday geographies of childrens popular culture," *Geoforum*, vol. 43, no. 1, pp. 4–13, 2012.

[3] Y.-H. Lin, "Integrating scenarios of video games into classroom instruction," in *Information Technologies and Applications in Education, 2007. ISITAE'07. First IEEE International Symposium on*. IEEE, 2007, pp. 593–596.

[4] S. M. Ogletree, C. N. Martinez, T. R. Turner, and B. Mason, "Pokémon: exploring the role of gender," *Sex roles*, vol. 50, no. 11-12, pp. 851–859, 2004.

[5] V. Vasquez, "What pokémon can teach us about learning and literacy," *Language Arts*, vol. 81, no. 2, pp. 118–125, 2003.

[6] S. D. Chen, "A crude analysis of twitch plays pokemon," *arXiv preprint arXiv:1408.4925*, 2014.

[7] D. Fudenberg and J. Tirole, *Game Theory*. MIT Press, 1991. [Online]. Available: https://books.google.co.jp/books?id=pFPHKwXro3QC

[8] H. Iida, K. Takahara, J. Nagashima, Y. Kajihara, and T. Hashimoto, "An application of game-refinement theory to mah jong," in *Entertainment Computing–ICEC 2004*. Springer, 2004, pp. 333–338.

[9] A. P. Sutiono, A. Purwarianti, and H. Iida, "A mathematical model of game refinement," in *Intelligent Technologies for Interactive Entertainment*. Springer, 2014, pp. 148–151.

[10] J. Bainbridge, "it is a pokémon world: The pokémon franchise and the

environment," *International Journal of Cultural Studies*, vol. 17, no. 4, pp. 399–414, 2014.

[11] G. Aloupis, E. D. Demaine, A. Guo, and G. Viglietta, "Classic nintendo games are (computationally) hard," *Theoretical Computer Science*, vol. 586, pp. 135–160, 2015.

[12] J. Tobin, *Pikachu's Global Adventure: The Rise and Fall of Pokémon*. Duke University Press, 2004. [Online]. Available: https://books.google.co.jp/books?id=U7hthImoc5AC

[13] Y.-H. Lin, "Pokemon: game play as multi-subject learning experience," in *Digital Game and Intelligent Toy Enhanced Learning, 2007. DIGI-TEL'07. The First IEEE International Workshop on*. IEEE, 2007, pp. 182–184.

[14] M. Moore, *Basics of Game Design*. CRC Press, 2016. [Online]. Available: https://books.google.co.jp/books?id=E9JG6JjPU-sC

[15] Pokemonshowdown, "An online pokemon battle simulator," http://pokemonshowdown.com, 2016.

[16] C. Panumate, S. Xiong, and H. Iida, "An approach to quantifying pokemon's entertainment impact with focus on battle," in *Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence (ACIT-CSI), 2015 3rd International Conference on*. IEEE, 2015, pp. 60–66.

[17] Bulbapedia, "An encyclopedia about pokemon," http://bulbapedia.bulbagarden.net/, 2016.

[18] Serebii, "Pokemon fan site," http://www.serebii.net/, 2016.

[19] W. Wig, *The Complete Pokemon Pokedex List (English Version)*. Gamas Publishing, 2016. [Online]. Available: https://books.google.co.jp/books?id=FOM0DAAAQBAJ

[20] Sweepercalc, "Pokemon usage stats," http://sweepercalc.com/stats/, 2016.