Dynamic game balancing implementation using adaptive algorithm in mobile-based Safari Indonesia game

Anny Yuniarti, Novita Nata Wardanie and Imam Kuswardayan

Department of Informatics, Faculty of Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

anny@if.its.ac.id

Abstract. In developing a game there is one method that should be applied to maintain the interest of players, namely dynamic game balancing. Dynamic game balancing is a process to match a player's playing style with the behaviour, attributes, and game environment. This study applies dynamic game balancing using adaptive algorithm in scrolling shooter game type called Safari Indonesia which developed using Unity. The game of this type is portrayed by a fighter aircraft character trying to defend itself from insistent enemy attacks. This classic game is chosen to implement adaptive algorithms because it has quite complex attributes to be developed using dynamic game balancing. Tests conducted by distributing questionnaires to a number of players indicate that this method managed to reduce frustration and increase the pleasure factor in playing.

1. Introduction

The easier it is for someone to create a game is one of the supporting factors that becomes the reason for the increasing number of mobile device games that continue to grow every day. Game developers must have innovations so that the game they create can be in demand by consumers and not quickly abandoned. Games generally have a level or difficulty level (easy, medium, and difficult) that can be played or selected directly by the user. On the average, the level of difficulty in traditional games runs linearly, in the sense that the higher the level, then, the higher the level of difficulty that will be faced by players. This level of difficulty already has its own set of attributes that the developer has set statically. Not infrequently the players considered a level in the game is too easy, thus making the people who play bored. Meanwhile, there is also the type of player who feels the game is too difficult to play, thus making the player feel frustrated. In the development of games that uses the concept of game balancing, players will be challenged based on their ability to play. Players are not forced to follow the flow of games that tend to increase the level of difficulty and make it frustrating. So it can be assumed in general, that the entertainment value of a balanced game is higher than games that tend to be too easy or too difficult [1].

The increase in game development industry, especially in Indonesia, is accompanied by the level of enthusiasm of the public on the ownership of smartphones, tablets, or other mobile devices. Many developers are increasingly motivated to create an exciting and value-added game for mobile device users. Therefore, this paper proposed a two-dimensional mobile game scrolling shooter type that introduces the Indonesian culture called Safari Indonesia, using dynamic game balancing as a method
to balance the game pattern of players with the complexity of the game so as to maintain the interest of players in playing.

2. Literature review

2.1. Dynamic game balancing
Dynamic game balancing or also called dynamic difficulty adaptation is one of the mechanisms in the game that aims to provide a good level design to challenge the players [1]. The level design is made as good as possible by the developer to keep the player interested in the game being played. Various characteristics of players in play, requires a developer to create a way for players to not feel bored or frustrated with the game played. Therefore, balancing the game is required. Balancing the game consists of changing the parameters, scenarios, and behavior of the game environment in order to prevent players from being bored in playing [2].

2.2. Adaptive algorithm
Adaptive algorithm is a collection of instructions to show that a function can adapt to changing circumstances or surrounding environments. Many examples can be given in terms of adaptive algorithm utilization, among others, on radar systems for false alarm rate detection that may occur, the utilization of adaptive algorithms is also done on machine learning process and optimization, data compression, and signal processing.

Adaptive ability has two interrelated meanings. First, it is easy to change. Second, adaptive means learning and transforming [3]. The first meaning means to adapt from one form to another, from one predetermined state to another predetermined state. Whereas in the second meaning, adaptation occurs when there is learning from the tests that have been done before, and there is a transformation to other circumstances that have not been known before. The adaptation can be applied to the game in several ways, among others are:
1. Player character
2. Non-player character on the game
3. Game environment
The easiest method to understand the three ways to adapt the game above is to change the difficulty level of non-player characters in the game. Non-player characters can be used to provide clues or as a support for players in realizing the needs and desires of play [4].

2.3. Player modelling
Player modeling becomes one of the simplest, flexible and lightweight approaches to be a solution in developing adaptive games [5]. The basic idea of game modeling is quite simple, that is, a game saves every player profile that contains abilities, weaknesses, skills, and other characters. As the game progresses, updates are made to the model according to the player's interaction with the game.

2.4. Scrolling shooter game
Scrolling shooter game, can also be called shoot 'em up, is a type of game in which a player controls a character, generally in the form of a fighter aircraft, to defend against enemy attacks in large numbers by shooting them. In this type of game scrolling shooter, players can control the character by moving it to a minimum of two directions (top down or right left) or to the four directions in general. In the game, not infrequently players also perform activities to collect power-ups used to develop character and defense.

3. Analysis and modelling

3.1. General description of the system
Safari Indonesia game to be built is a scrolling shooter manifest game in which there is a single player who try to defend themselves from enemy attacks that continue to come periodically. The game will be developed in two modes, adventure and endless. The adventure mode is a game developed with static level configuration, while the endless mode is a mode developed by implementing dynamic game balancing using adaptive algorithm.

### 3.2. Functional requirements

Functional requirements are a description of the information that occurs on the system to be built. Safari Indonesia game apps have four functional requirements as outlined in Table 1.

| Functional ID | Functional requirements                  |
|---------------|-----------------------------------------|
| F-01          | Play the game with endless mode         |
| F-02          | Play the game with adventure mode       |
| F-03          | Perform character updates               |
| F-04          | View the status of the game             |

### 3.3. Player modelling design

The main part of the player modeling system is the model itself. The model is actually a collection of numerical attributes or traits that describe the playing style of each player [5]. The player modeling discussed in this section focuses on the design of endless modes that implement dynamic game balancing using adaptive algorithms.

In modeling and matching the difficulty level of the game with player game patterns, enemies in endless fashion games have behaviors that are compiled into a variable hereinafter called $v$. Behavior referred to in variable $v$, consists of:

\[
\begin{align*}
\text{speed} &= \text{speed}_0 \\
\text{shotIntensity} &= \text{shotIntensity}_0 \\
\text{enemyNumbers} &= \text{enemyNumbers}_0
\end{align*}
\]

This game has three levels of difficulty which hereinafter called player type.

\[
\text{playerType} = \{\text{easy, medium, difficult}\}
\]

Furthermore, each type of player that has a constant multiplier $p$ is useful for matching characters to be done later. Multipliers for each type are set as follows:

\[
\begin{align*}
p(\text{easy}) &= 0.85 \\
p(\text{medium}) &= 1.05 \\
p(\text{difficult}) &= 1.3
\end{align*}
\]

The next design is modeling for player characters that will be observed against it. Player character modeling is implemented in the form of numeric numbers ranging from 0 to 1 as shown in Table 2.

When a player starts the game in an endless fashion, the player will be initialized with a moderate player type and has a player modeling attribute value that is in the range of medium player type minimal value up to the maximum value of the medium player type. The initialization values are performed with equation (4).

After getting the middle value of each character, the values are summed to the player's initial performance when starting the game. The equation for summing the middle values of each character is shown in equation (5).
Table 2. Player modeling.

| Player character                | Easy   | Medium | Hard  |
|--------------------------------|--------|--------|-------|
|                                | Min    | Max    | Min   | Max   | Min   | Max   |
| Accuracy                       | 0.0    | 0.2    | 0.2   | 0.6   | 0.6   | 1.0   |
| Health variation               | 0.6    | 1.0    | 0.3   | 0.6   | 0.0   | 0.3   |
| Number of defeated enemies     | 0.0    | 0.3    | 0.3   | 0.7   | 0.7   | 1.0   |
| Total                          | 0.6    | 1.5    | 0.8   | 1.9   | 1.3   | 2.3   |

\[
\text{Character}_i = \frac{\text{CharacterMinimal}_i + \text{CharacterMaximum}}{2}
\]

(4)

\[
\text{Player's initial performance} = \sum_{i=1}^{n} \text{Character}_i
\]

(5)

3.4. Adaptive algorithm of Safari Indonesia game

The design of the algorithm formulated in this paper is shown in pseudocode Figure 1. The learning rate constant used in this algorithm has a range of values 0 - 1. The smaller the value, it takes more attributes to change the player model. The range of learning rate constants recommended by reference [5] is 0.1 - 0.3.

\begin{verbatim}
\text{a} \leftarrow \text{learning rate} \\
\text{playerType}_0 \leftarrow \text{player type initialization} \\
\text{cl} = \frac{\text{CharacterMinimal}_i + \text{CharacterMaximum}}{2} \text{(The number of i characters becomes the player performance initialization)} \\
\text{while (true)} \\
\hspace{1em} \text{count the observed character (ciobs)} \\
\hspace{2em} \text{cl} \leftarrow \text{cl} + \alpha \times (\text{ciobs} - \text{cl}) \\
\hspace{2em} \text{performance} = \sum_{i=1}^{n} \text{ci} \\
\hspace{2em} \text{if performance} \in [\text{MIN}^{\text{new}}, \text{MAX}^{\text{new}}] \text{ then} \\
\hspace{3em} \text{playerType[level]} \leftarrow \text{playerType()}} \\
\hspace{2em} \text{else if playerType[level] } \neq \text{playerType[level - 1]} \text{ then} \\
\hspace{3em} v \leftarrow \text{match(playerType[level])} \\
\hspace{2em} \text{else} \\
\hspace{3em} \text{maintain player behavior} \\
\hspace{2em} \text{end if} \\
\hspace{2em} \text{save the performance of each level} \\
\hspace{2em} \text{end if} \\
\hspace{1em} \text{end while}
\end{verbatim}

Figure 1. The adaptive algorithm pseudocode.

4. Testing

4.1. Game control test

Game control is an important element in the game that can support the user's comfort in playing the game. The results of the game control tests in Figures 2 and 3 show a player who moves the game characters of an aircraft in all directions to fire a bullet to the enemy. This shows that the implementation of game controls in Safari Indonesia Games runs well.
Figure 2. Game control testing in endless mode.

Figure 3. Game control testing in adventure mode.

Table 3. Player performance test results.

| Level | Performance | Type | Accuracy | Live | Defeated enemy |
|-------|-------------|------|----------|------|----------------|
| 1     | 1.3500      | 2    | 0.29     | 0.33 | 0.29           |
| 2     | 1.2626      | 1    | 0.51     | 0.67 | 0.50           |
| 3     | 1.3452      | 1    | 1.00     | 1.00 | 1.00           |
| 4     | 1.6762      | 2    | 1.00     | 0.71 | 1.00           |
| 5     | 1.8838      | 2    | 0.98     | 1.00 | 0.96           |
| 6     | 2.1070      | 3    | 0.96     | 0.63 | 0.91           |
| 7     | 2.1849      | 3    | 0.92     | 0.67 | 0.87           |
| 8     | 2.2394      | 3    | 0.72     | 0.50 | 0.77           |
| 9     | 2.1188      | 3    | 0.68     | 0.36 | 0.69           |
| 10    | 2.0983      | 3    | 0.50     | 0.68 | 0.56           |
4.2. **Performance Test**

Performance test results in the calculation of attributes that characterize the player model and the resulting performance of the sum of the attributes. Performance is what will determine the type of player.

The experiments shown in Table 3 indicate that the player has decreased performance with a value of 1.2626 at level 2 which causes the player type to change in type 1. The player successfully improves the performance on the next level, and changes the type of player in type 2 at level 4. Next, players experience increased performance that causes players to change to type 3 at level 6.

Experiments to test the performance of players in Safari Indonesia Game prove that the implementation of adaptive algorithms applied to balance the ability of players with difficulty level of the game is declared successful (Figure 4 and 5).

4.3. **Functional requirements test**

The functional requirements test of the game is performed by the black box method in which testing is performed by simply observing the results of running programs and checking each use case that has been implemented in the game. Table 4 is a breakdown of the results of functional requirements testing.

4.4. **User Testing**
User testing is done to test important factors in the game. Questionnaires and factors tested in the questionnaire analysis refer to reference [6]. Table 5 shows the relationship between the factors tested in the game with the statement on the questionnaire.

**Table 4.** Functional requirement test result.

| Functional ID | Functional requirements                      | Test result |
|---------------|---------------------------------------------|-------------|
| F-01          | Play the game with endless mode             | Successful  |
| F-02          | Play the game with adventure mode           | Successful  |
| F-03          | Perform character updates                   | Successful  |
| F-04          | View the status of the game                 | Successful  |

**Table 5.** Relationship between important factors in the game with statements on the questionnaire.

| Questionnaire number | Factors                                      |
|----------------------|----------------------------------------------|
| 1, 4, 5              | Fun                                          |
| 2, 3                 | Frustration                                  |
| 6 – 38               | Important elements in the game               |
| 6 – 12, 25, 28       | Game controls                                |
| 13 – 18              | Game facilities                              |
| 26 – 31              | Game environment                             |
| 32 – 37, 39<sup>a</sup> | Rules                                       |

<sup>a</sup> Statement of questionnaire number 39 is only intended for endless fashion games.

**Table 6.** Safari Indonesia game correspondent details.

| Correspondent | Total | Men | Women | Beginner | Skilled |
|---------------|-------|-----|-------|----------|---------|
|               | 18    | 8   | 10    | 8        | 10      |
| Percentage    | 44%   | 56% | 44%   | 56%      |

**Table 7.** Results of questionnaire analysis of beginner players.

| Factor               | Endless Mode | Adventure Mode | Deviation (%) |
|----------------------|--------------|----------------|---------------|
|                      | Total        | Average        | Total         | Average      |          |
| Fun                  | 130          | 5.4166         | 120           | 5            | 8        |
| Frustration          | 51           | 3.1875         | 51            | 3.1875       | 0        |
| Important elements   | 1249         | 4.7310         | 1216          | 4.6060       | 2.67     |
| Game controls        | 337          | 4.6805         | 334           | 4.6388       | 0.89     |
| Game facilities      | 208          | 4.4583         | 196           | 4.0833       | 8.78     |
| Game environment     | 240          | 5              | 241           | 4.4629       | 11.35    |
| Rules                | 284          | 5.1428         | 229           | 4.0892       | 22.82    |
Table 8. Comparison of playing duration.

| Player ID | Player Type | Playing Duration (Minute) |
|-----------|-------------|----------------------------|
|           |             | Adventure Mode | Endless Mode |
| 1         | Advanced    | 5             | 10           |
| 2         | Advanced    | 4             | 19           |
| 3         | Advanced    | 5             | 6            |
| 4         | Beginner    | 4             | 19           |
| 5         | Beginner    | 5             | 13           |
| 6         | Advanced    | 6             | 11           |
| 7         | Beginner    | 4             | 11           |
| 8         | Beginner    | 5             | 16           |
| 9         | Beginner    | 4             | 5            |
| 10        | Advanced    | 6             | 10           |
| 11        | Advanced    | 5             | 13           |
| 12        | Beginner    | 3             | 10           |
| 13        | Advanced    | 7             | 13           |
| 14        | Advanced    | 7             | 6            |
| 15        | Advanced    | 3             | 8            |
| 16        | Advanced    | 5             | 16           |
| 17        | Beginner    | 4             | 13           |
| 18        | Beginner    | 5             | 8            |
|           | Average     | 5             | 12           |

User testing of the Safari Indonesia game was conducted by 18 correspondents. Those 18 people consists of 8 men and 10 women with the age range between 20-22 years. 18 correspondents are divided into two groups of players, advanced and beginners based on the experience of playing on a similar game (Table 6). Testing performed by users with a beginner type player for both endless and adventure modes can be seen in Table 7.

The results of the analysis of the questionnaires filled by beginner players show that the level of fun in endless mode game that applied the dynamic game balancing method using adaptive algorithm has a higher value than the adventure mode game, while the frustration rate of both modes has the same value. The average value of other tested factors such as important elements in the game, game controls, game facilities, game environments, and rules show higher points for endless modes.

Next, the results of the questionnaire analysis of the advanced players show the level of fun in the endless mode is still higher than the adventure mode. However, the frustration rate in endless mode games has significantly lower value than the adventure mode. This indicates that the player can feel frustrated if the level of difficulty at a level does not match with his ability, causing the players to stop to play. Other factors analyzed in the questionnaire showed higher values in endless mode, except for one factor that is an endless mode game facility that has a lower value than the adventure mode.

Table 8 shows the result of comparison of level completion by each correspondent in endless and adventure mode. The results of the final recapitulation of all correspondents show a higher value in the endless mode. This proves that players can last longer to play a game that implements the dynamic game balancing method in it.
5. Conclusions
From the process of working on this research which through the stage of design, implementation, and testing, the following conclusions are obtained:
1. Safari Indonesia game successfully implemented the rules of the game of scrolling shooter type games well.
2. Game controls for Safari Indonesia games have been tailored to the requirements of mobile devices to make it easier for players to play on the Android platform.
3. Indonesia Safari game that has implemented the concept of dynamic game balancing using adaptive algorithm, successfully balancing between player game pattern as measured by accuracy, number of defeated enemies, and amount of health left with game environment especially enemy characters implemented on speed behavior, shot intensity, and number of enemies at each level.
4. Based on the user testing process, endless mode has a higher level of fun than adventure mode. According to an advanced player endless mode also has a low frustration value. This shows the achievement of the goal of the implementation of dynamic game balancing which is to reduce the frustration and boredom felt by players when playing a game.

References
[1] Andrade G, Ramalho G, Gomes A Sandro and Corruble V 2006 AAAI conference on artificial intelligence and interactive digital entertainment (California), p 3-8
[2] Koster R 2004 Theory of Fun for Game Design (Phoenix: Paraglyph Press)
[3] Charles D and Black M 2004 Proc. of the Int. Conf. on Computer Games: Artificial Intelligence, Design and Education, p 29-35
[4] Charles D, Kerr A, McNeill M, McAlister M, Black M, Kcklich J, Moore A and Stringer K 2005 Proc. of the digital games research conference, p 100
[5] Houlette R 2004 Player modeling for adaptive games AI Game Programming Wisdom 2
[6] Calvillo-Gámez E H, Cairms P and Cox AL 2010 Assessing the core elements of the gaming experience Evaluating User Experience in Games: Human-Computer Interaction Series ed Bernhaupt R (London: Springer)